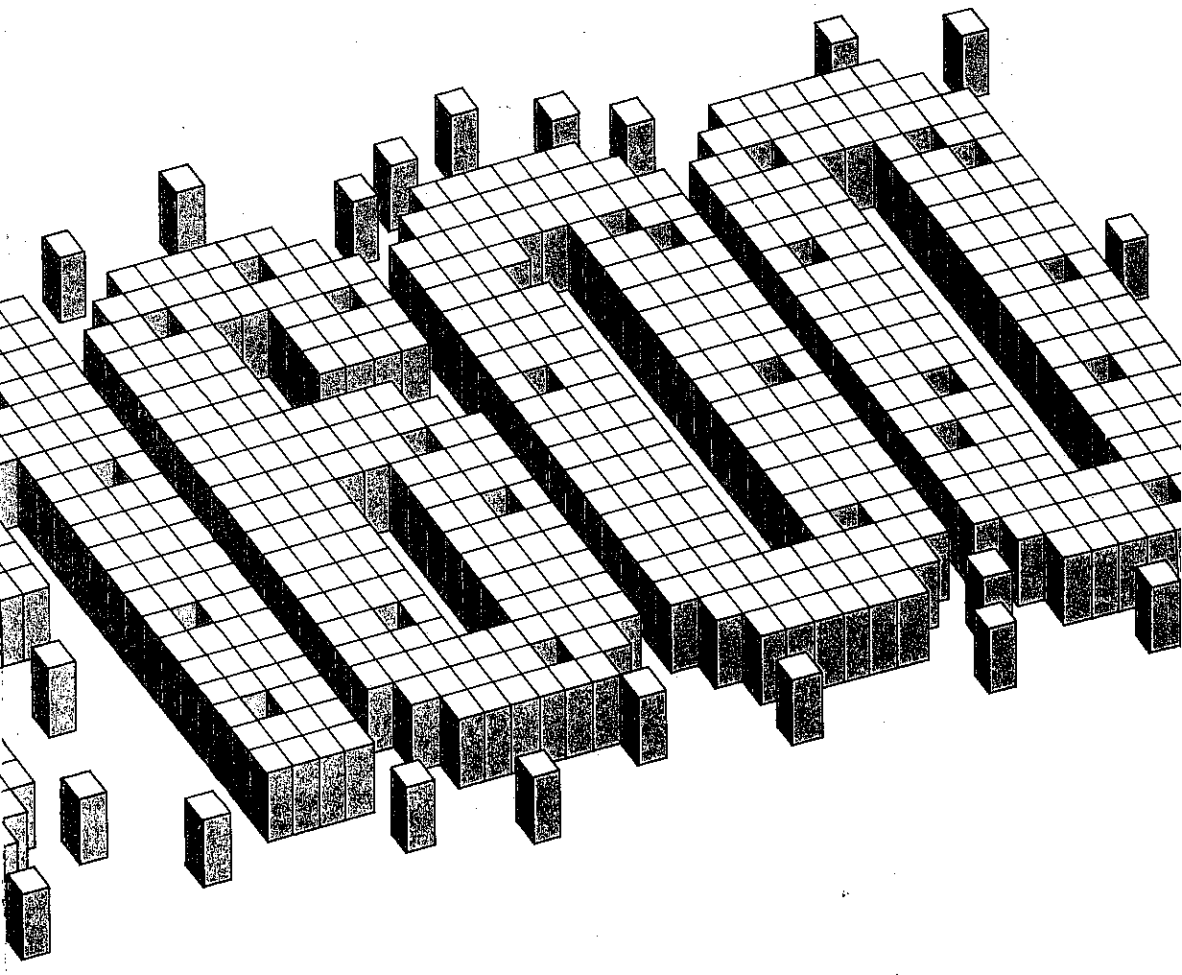
SHARP®

ASCHENCOMPUTER

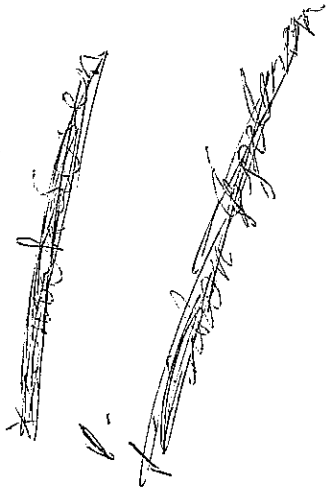
MODELL

PC-1600

BEDIENUNGSANLEITUNG

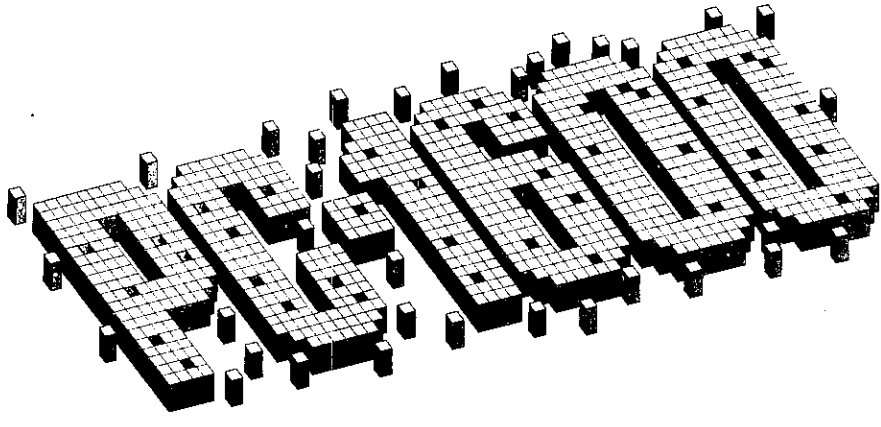


V unrichtig
V richtig



SHARP®

TASCHENCOMPUTER
MODELL



BEDIENUNGSANLEITUNG

HQE 5/86

~~3-1~~

2-4 ✓✓

3-2 ✓✓

3-3 ✓✓

3-4 ✓✓

3-10 ✓✓

3-11 ✓✓

INHALTSVERZEICHNIS

EINLEITUNG 6

Ausstattungsmerkmale 6
Gliederung dieses Handbuches 7

TEIL I VORBEREITUNGEN .....

1. Beschreibung des PC-1600 1-1

1.1 Hinweise zum Auspacken 1-1
1.2 Hardware-Überblick 1-1
1.3 Die Tastatur des PC-1600 1-4
1.4 Anmerkungen zur Software..... 1-7

TEIL II INBETRIEBNAHME UND GEBRAUCH DES PC-1600 .....

2. Stromversorgung des PC-1600 2-1

2.1 Einsetzen der Batterien 2-1
2.2 Überwachung der Batterien 2-2
2.3 Anschluß des Netzadapters 2-3

3. Ein- und Ausschalten des Computers 3-1

3.1 Einschaltung des PC-1600 3-1
3.2 Ausschaltung des PC-1600 3-5
3.3 RESET-Möglichkeiten 3-6

4. Das PC-1600-Display 4-1

4.1 Anzeige-Modi 4-1
4.2 Einstellung des Kontrastes 4-1
4.3 Symbole der Statuszeile 4-2
4.4 Wahl der Betriebsarten 4-4
4.5 Editierende Tastenfunktionen 4-5
4.6 Setzen von Uhrzeit und Datum 4-7

5. Rechnen mit dem PC-1600 5-1

5.1 Wahl der Betriebsart 5-1
5.2 Tasten-Funktionen 5-2
5.3 Rechenbeispiele 5-3
5.4 Kettenrechnungen 5-4
5.5 Rückholung eines Rechenausdrucks 5-5
5.6 ERROR-Codes 5-6
5.7 "Arithmetische" BASIC-Funktionen 5-7

INHALTSVERZEICHNIS

TEIL III BENUTZUNG DER SCHNITTSTELLEN UND OPTIONEN .....

6. Erweiterung des PC-1600 6-1

6.1 System-Überblick 6-1

6.2 Erweiterung durch RAM-Module 6-3
a) Tabelle der RAM-Module 6-4
b) Hinweise zur Benutzung der Module 6-4
c) RAM-Disk-Befehle 6-6
d) ERROR-Codes 6-8

6.3 Serielle Schnittstellen 6-8

A) RS-232C-Schnittstelle 6-8
a) Hinweise zur Benutzung 6-9
b) Standardmäßige Einstellungen 6-10
c) RS-232C-Anschlußbelegung 6-11
d) RS-232C-Befehle 6-12
e) ERROR-Codes 6-13

B) Serieller optischer Port (SIO) 6-13
a) Hinweise zur Benutzung 6-14
b) Standardeinstellungen 6-15
c) Anschlußbelegung 6-15
d) Befehle für das SIO-Interface 6-16
e) ERROR-Codes 6-17

6.4 Analoger-Eingabe-Port 6-17
a) Elektrische Kennwerte 6-18
b) Befehle für den analogen Eingang 6-18

6.5 Drucker mit Recorder-Interface 6-19
a) Hinweise zum Gebrauch 6-20
b) Befehle für den Drucker 6-21
c) ERROR-Codes 6-22

6.6 Gebrauch eines Cassetten-Recorders 6-22
a) Hinweise zum Gebrauch 6-23
b) Recorder-Befehle 6-25
c) ERROR-Codes 6-25

6.7 Disketten-Laufwerk 6-26
a) Hinweise zum Gebrauch 6-27
b) Disketten-Befehle 6-29
c) ERROR-Codes 6-31

7. Benutzung der PC-1500-Optionen 7-1

TEIL IV BASIC-REFERENZTEIL .....

8. Grundlegendes zur Programmierung 8-1

8.1 Eingabe von BASIC-Befehlen 8-1
8.2 Start und Ablauf eines Programmes 8-3
8.3 Speicherbelegung 8-5

9. Betriebsarten unter BASIC .....

9.1 Tastenfunktionen 9-1
A) Alphanumerische Tasten 9-1
B) Umschalt-Tasten 9-1
C) Spezial-Tasten 9-3

9.2 Anzeige-Modi 9-6

9.3 Editier-Modus 9-7

9.4 RESERVE-Modus 9-9
A) Auswahl der RESERVE-Modi 9-10
B) Setzen der Funktionstasten-Strings 9-10
C) Abruf der Tasten-Belegungen 9-11
D) Funktionstasten-Menüs 9-12
E) Speichern und Laden
von Funktionstasten-Belegungen 9-13
F) Vordefinierte Tastaturbelegung 9-13

10. Datendarstellung 10-1

10.1 Daten-Typen 10-2
A) Textdaten und Zeichensätze 10-2
B) Zeichen-Strings 10-3
C) Numerische Daten 10-3

10.2 Konstanten 10-4
A) String-Konstante 10-4
B) Numerische Konstante 10-4

10.3 Variablen..... 10-5
A) Variablen-Namen 10-6
B) Variablen-Typen 10-7
a) Numerische Standard-Variablen .. 10-7
b) Einfache numerische Variablen .. 10-8
c) Numerische Felder (Arrays) 10-8
d) Standard-String-Variablen 10-10
e) Einfache String-Variablen 10-10
f) String-Arrays 10-10

10.4 Ausdrücke und Operatoren 10-11
a) Arithmetische Operatoren 10-11
b) Vergleichsoperatoren 10-12
c) Logische Operatoren 10-13
d) Funktionale Operatoren 10-15

11. Dateien 11-1

11.1 Datei-Bezeichner 11-1
A) Logischer Gerätename (Medium) 11-2
B) Dateiname 11-2
C) Extension 11-2

11.2 Speichern und Laden von Dateien 11-3
A) Cassetten-Dateien 11-3
B) Disketten-Dateien 11-4
a) Inhaltsverzeichnisse 11-4
b) Anzahl der Disketten-Dateien ... 11-4
c) Mehrdeutige Datei-Bezeichner ... 11-5

11.3 Schreibschutz 11-6

11.4 Erstellung einer Datei 11-6

11.5 Zugriff auf Dateien 11-9

11.6 Änderung einer Datei 11-10

12. Zugriff auf die seriellen Ports 12-1

12.1 Auswahl der Schnittstelle 12-1

12.2 Protokoll-Optionen 12-2

12.3 Kommunikations-Parameter 12-2

12.4 Empfangs-Puffer 12-3

12.5 Ausgabe über einen seriellen Port 12-3
A) Senden von Programmen und Daten ... 12-3
B) Senden von Programm-Dateien 12-4
C) Senden von Steuer-Codes 12-4

12.6 Empfang über einen seriellen Port 12-5
A) Empfang von Daten 12-5
B) Empfang von Dateien 12-5

13. Fehler-Korrektur 13-1

13.1 Syntax-Fehler 13-1
13.2 Trace-Modus 13-2
13.3 Fehlerbehandlungs-Routinen 13-2

14. Erklärung der BASIC-Befehle 14-1

14.1 Einleitung 14-1
14.2 Begriffsdefinitionen 14-1
14.3 Befehls-Kürzel 14-2
14.4 Anwendungssymbolik 14-3

TEIL V ANHANG .....

A. Auswechseln der Batterien A-1
B. Austausch der RAM-Module A-3
C. Tabellen der Zeichen-Codes A-4
D. Speicher-Aufteilung A-6
E. Maschinensprache-Programme A-9
F. ERROR-Codes des PC-1600 ..... A-11
Programm-Fehler A-11
Recorder-Fehler A-15
Drucker-Fehler A-15
Port-Fehler A-16
Datei-Fehler A-16
G. Liste der BASIC-Befehle A-18
H. Kompatibilität zum PC-1500 und dessen Optionen A-22
I. Vorsichtsmaßnahmen und Problemlösungshinweise A-25
J. Technische Spezifikationen A-28
K. Syntax-Diagramme A-30

REGISTER .....

EINLEITUNG

Herzlichen Glückwunsch zum Kauf des SHARP PC-1600 Computers.

Der Taschencomputer PC-1600 stellt den neuesten Stand hinsichtlich Universalität und Portabilität dar. Er paßt zwar gerade in eine Handfläche, bietet aber die Leistung, wie man sie sonst nur von größeren Computern her gewohnt ist. Mit einer Reihe speziell entwickelter Optionen (gesondert erhältliches Zubehör) kann der PC-1600 leicht zu einem kompakten System erweitert werden, das auch Ihren besonderen Anforderungen gerecht wird.

Austattungsmerkmale

Hier sind einige der Ausstattungsmerkmale, die Ihnen zeigen, daß es sowohl für den erfahrenen Anwender als auch für den Neuling keine Alternative zum PC-1600 gibt:

- Leistungsfähiger 8-Bit-CMOS-Prozessor (Z-80 kompatibel).
- Standard-RAM von 16KByte Speicherkapazität, die auf fast 80KByte aufgerüstet werden kann.
- Vierzeilige Flüssigkristallanzeige (LCD) mit 26 Zeichen pro Zeile und einer Darstellung mittels 5x7-Punktmatrix
- Alle 156x32 Punkte der Anzeige sind einzeln ansteuerbar.
- Echtzeit-Uhr unterstützt Weck- und Alarm-Funktionen.

Ihr PC-1600 arbeitet mit einer speziell angepaßten BASIC-Version die im Speicher fest verdrahtet ist und Ihnen das Schreiben von eigenen Programmen auf einfache und schnelle Weise ermöglicht.

Der PC-1600 ist batteriebetrieben, damit er auch unterwegs und überall genutzt werden kann. Zu Hause oder im Büro kann er zur Entlastung der Batterien aber auch über einen Netzadapter vom Wechselstromnetz versorgt werden.

Über folgende drei Schnittstellen (Interfaces, Ports) können Sie den PC-1600 direkt an externe Ein-/Ausgabegeräte (I/O devices) anschließen:

RS-232C-Interface,
SIO-Interface (optoelektronische Schnittstelle),
Analog-Port (analoger Eingang).

Der PC-1600 kann mit folgendem Zubehör betrieben werden:

- o CE-1600M 32KByte-RAM-Modul mit Batterie-Pufferung zur Programmspeicherung und Speichererweiterung.

- o CE-1600P Drucker zum vierfarbigen Ausdruck von Text und Grafiken. Über diesen Drucker besteht die Anschlußmöglichkeit eines Cassetten-Recorders zur Speicherung von Programmen.

- o CE-1600F Disketten-Laufwerk für zweiseitig beschreibbare 2,5"-Disketten, die jeweils 64 KByte pro Seite speichern können.

Wegen der Kompatibilität des PC-1600 zu seinem Vorgänger, dem PC-1500, kann er auch an die PC-1500-Peripherie angeschlossen werden. Diese Kompatibilität bezieht sich auf den Drucker, den Cassetten-Recorder, die RAM-Module, die Schnittstellen und als wichtigstes: auch auf die Programme.

Gliederung dieses Handbuches

Um Ihnen die Benutzung dieses Handbuches zu erleichtern, haben wir es in folgende fünf Teile gegliedert:

TEIL I (VORBEREITUNGEN)

gibt eine grundlegende Beschreibung des PC-1600, um Sie mit seiner Ausstattung und seinen Besonderheiten vertraut zu machen. Diesen Abschnitt sollte jeder Leser als "Muß" betrachten.

TEIL II (INBETRIEBNAHME UND GEBRAUCH DES PC-1600)

beginnt bei der Inbetriebnahme des PC-1600 inklusive dem Einsetzen der Batterien, Ein- und Ausschalten des Computers sowie Setzen von Uhrzeit und Datum der eingebauten Echtzeit-Uhr. Ebenso zeigt dieser Teil, wie man eingegebene Daten mit den Editiertasten korrigiert und schließt mit einem Kapitel über die Verwendung des PC-1600 als Taschenrechner ab.

TEIL III (BENUTZUNG DER SCHNITTSTELLEN UND OPTIONEN)

beschreibt die drei Schnittstellen des PC-1600 sowie seine Modulfächer. Er liefert Informationen, wie der Drucker, das Diskettenlaufwerk und ein Cassetten-Recorder zu verwenden sind. Im letzten Kapitel wird außerdem auf die Kompatibilität zu den Optionen des PC-1500 eingegangen.

TEIL IV (BASIC-REFERENZTEIL)

beschreibt das "Betriebssystem" Ihres Computers, die fest eingebaute Programmiersprache: PC-1600-BASIC. Er gibt eine Anleitung zum Schreiben eigener Programme sowie zur Steuerung der optionalen Zusatzgeräte. Dieses Kapitel beschreibt das grundsätzliche Konzept des BASICs und die Anwendung dieser Programmiersprache auf dem PC-1600. Es kann und will aber kein Lehrbuch für BASIC-Anfänger sein. Diesbezüglich gibt es umfangreiche und gute Literatur im einschlägigen Buchhandel. Den Abschluß von Teil IV bildet ein komplettes Wörterbuch der BASIC-Befehle, wie sie vom PC-1600 verstanden und mit diesem genutzt werden können.

TEIL V (ANHANG)

beschreibt das Auswechseln der Batterien und der RAM-Module. Außerdem finden Sie hier eine Zusammenstellung aller BASIC-Befehle und ERROR-Codes (Fehlermeldungen). Desweitern gibt dieser Teil Auskunft über technische Spezifikationen und die Aufteilung des Speichers. Auch der Zeichensatz und die Grundlagen der Syntaxdiagramme sind hier erläutert.

Wir schlagen vor, sich mit diesem Handbuch vertraut zu machen, bevor Sie den Computer einschalten und mit ihm arbeiten möchten. Damit können unnötige Probleme, die durch eine Fehlbedienung entstehen, vermieden werden. Lesen Sie daher bitte unbedingt den TEIL I (VORBEREITUNGEN) dieses Handbuches, der die besonderen Merkmale des PC-1600 beschreibt. Alle weiteren Teile können je nach Bedarf gelesen werden und als Nachschlagewerk dienen.

VORBEREITUNGEN

Mit einer Gesamtbeschreibung des PC-1600 wollen wir diesen Teil beginnen, nachdem Sie ihn ausgepackt und sein Zubehör auf Vollständigkeit überprüft haben.

Danach erfolgt eine Einführung in die Besonderheiten von Tastatur und Tastenfunktionen des PC-1600.

Den Schluß bildet eine Kurz-Beschreibung der eingebauten PC-1600-Software.

KORREKTUR-BLATT

Während der Erstellung des Bedienungs-Handbuches sind am PC-1600 einige technische Änderungen vorgenommen worden, die wir wegen des bereits angelaufenen Drucks nicht mehr in die Beschreibungen haben einfließen lassen können. Wir bitten Sie deshalb um Ihr geschätztes Verständnis, daß wir Sie in Form eines Korrektur-Blattes auf diese Änderungen hinweisen.

1) Speicherumfang des Anwenderbereiches

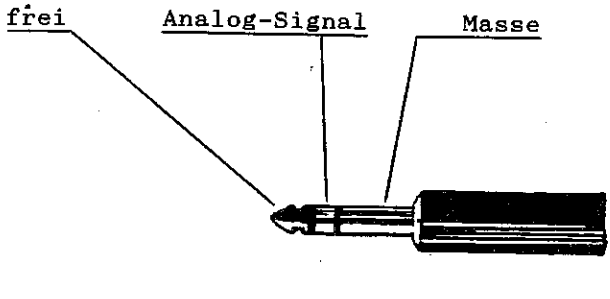
Aufgrund dieser Änderungen benötigt der PC-1600 für die system-spezifischen Belange etwas mehr RAM. Dem Anwender stehen deshalb anstatt der erwähnten 12090 Bytes maximal 11834 Bytes zur freien Verfügung.

2) Anschluß eines Barcode-Lesers

Der Barcode-Leser wird nicht (wie bisher beschrieben) an den Analog-Eingang angeschlossen, sondern an das SIO-Interface.

3) Anschlußbelegung des Analog-Eingangs

Die Anschlußbelegung des 3,5"-Klinkensteckers für den analogen Eingang sieht folgendermaßen aus:



1. Beschreibung des PC-1600

Wir wissen, daß Sie am liebsten so bald wie möglich mit Ihrem neuen PC-1600 arbeiten wollen. Um ihn jedoch von Beginn an sachgemäß bedienen und damit künftige Probleme vermeiden zu können, sollten Sie sich einige Minuten Zeit nehmen und nachfolgende Seiten beachten, bevor Sie zu den anderen Abschnitten dieses Handbuches übergehen.

1.1 Hinweise zum Auspacken

Nachdem Sie Ihren PC-1600 und sein Zubehör ausgepackt haben, überprüfen Sie bitte nochmals den Packungsinhalt anhand der untenstehenden Liste auf Vollständigkeit. Sollte irgendetwas fehlen oder einen Schaden aufweisen, setzen Sie sich bitte unverzüglich mit Ihrem SHARP-Händler in Verbindung.

Ihre PC-1600-Packung sollte folgenden Inhalt aufweisen:

- Den SHARP PC-1600 Taschen-Computer
- Ein Aufbewahrungs-Etui
- Zwei Tastatur-Schablonen
- Vier Mignonzellen (Typ AA)
- Dieses Bedienungs-Handbuch

Sofern alles vorhanden und unbeschädigt ist, legen Sie bitte das gesamte Verpackungsmaterial in den Karton zurück und bewahren es für einen eventuellen späteren Gebrauch auf. Diese Maßnahme erweist sich als sinnvoll, falls Sie den PC-1600 einmal versenden oder für längere Zeit ungenutzt aufbewahren wollen.

1.2 Hardware-Überblick

Der PC-1600 besitzt eine QWERTY-Tastatur, wie man sie in ähnlicher Form von einer Schreibmaschine her gewohnt ist, eine Flüssigkristallanzeige (LCD), zwei Modul-Schächte für Speichererweiterungen und drei Anschlüsse für verschiedene optionale Peripherie-Geräte.

Die folgenden Seiten beschreiben die einzelnen Teile des PC-1600, um Sie mit deren Lage und dessen Funktionen vertraut zu machen.

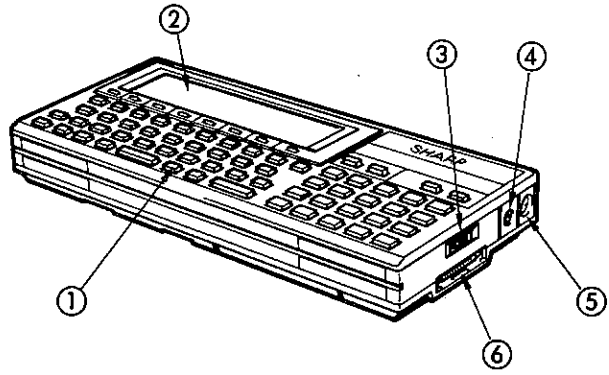


Abbildung 1 : Vorderansicht des PC-1600

① Tastatur

Mit seiner schreibmaschinenähnlichen Tastatur und dem numerischen Tastenfeld besitzt der PC-1600 insgesamt 69 Tasten, die teilweise speziellen Zwecken dienen oder programmierbar sind.

② LCD-Anzeige

Die Anzeige des PC-1600 besteht aus vier Zeilen, in denen jeweils 26 Zeichen dargestellt werden können. Zur bequemen Ablesung ist der Kontrast einstellbar.

③ Kontrast-Einsteller

Wenn sich die Arbeitsposition des Computers oder die umgebenden Lichtverhältnisse ändern, kann mit diesem Einsteller der Kontrast für eine bestmögliche Ablesung verändert werden.

④ Analog-Port

Dieser Anschluß erlaubt dem PC-1600, analoge Daten entgegenzunehmen, wie sie beispielsweise von Barcode-Lesern oder Sensoren geliefert werden.

⑤ Optisches Interface

Benutzen Sie diesen mit SIO bezeichneten Anschluß, um mittels Lichtleiterkabel mit externen Geräten bei höchster Geschwindigkeit störungsfrei kommunizieren zu können.

⑥ RS-232C-Interface

Über diesen Anschluß können Daten seriell gesendet oder empfangen werden. Dieses ist der Standard-Anschluß für andere Geräte, wie Computer, Modems oder Drucker.

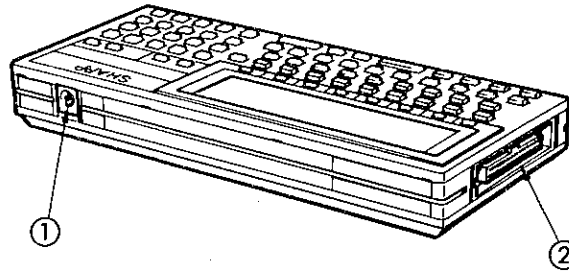


Abbildung 2 : Rückansicht des PC-1600

① Adapter-Anschluß

Wird der PC-1600 stationär betrieben, so kann an diese Buchse ein Netzadapter (EA-160 oder EA-150) angeschlossen werden und somit die Speisung des Computers über das Wechselstromnetz erfolgen. Das schont die Batterien und verlängert deren Lebensdauer.

② System-Bus

Dieser Bus-Anschluß stellt eine direkte Verbindung zum Herz des Computers, dem Mikroprozessor, her. Seine Daten und Steuersignale lassen sich hierüber an speziell auf den Taschencomputer PC-1600 (bzw. seinen Vorgänger: PC-1500) zugeschnittene Geräte, wie optionale Drucker oder Interface-Einheiten und dergleichen mehr anschließen.

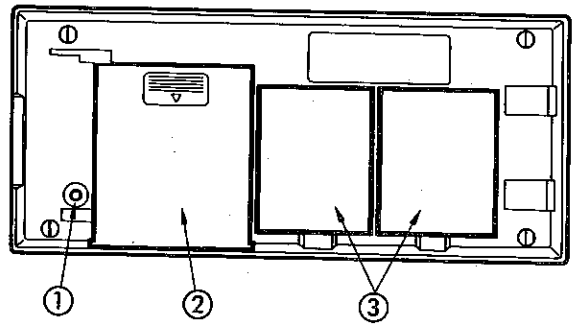


Abbildung 3 : PC-1600 von unten gesehen

① **Reset-Schalter**

Sollte der PC-1600 durch besondere Umstände einmal die Kontrolle über sich verlieren, sich also "aufgehängt" haben, kann er mit Hilfe dieses Schalters wieder in den normalen Betriebszustand zurückversetzt werden, sofern die Bedienung der [ON] - und [OFF] - Tasten versagt.

ACHTUNG: Es ist nicht gewährleistet, daß Daten und Programme nach Betätigung der Reset-Taste erhalten bleiben. (Siehe Kapitel 3: Reset des Computers).

② **Batterie-Fach**

Wird der PC-1600 nicht über einen Netzadapter am Wechselstromnetz betrieben, versorgen ihn vier Batterien (Mignonzellen) mit der notwendigen Energie (siehe Kapitel 2: Stromversorgung).

③ **Modul-Fächer**

Zur Erweiterung des internen Arbeitspeichers oder der schnellen Bereitstellung von Programmen lassen sich in diese Fächer RAM-Module verschiedenster Art einsetzen. Diese können als Speicher-Module, Programm-Module oder RAM-Disks verwendet werden.

1.3 Die Tastatur des PC-1600

Dieser Abschnitt weist auf einige wichtige Merkmale der PC-1600-Tastatur hin. Auf Tastenfunktionen, die sich auf das eingebaute BASIC beziehen, wird in Kapitel 9 eingegangen. Die Anwendung der Taste [CTRL] und das Editiern von Eingaben sind in Kapitel 4 erläutert.

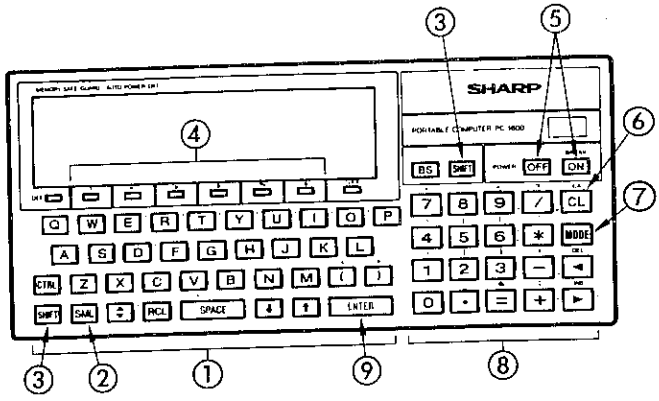


Abbildung 4 : Die Tastatur des PC-1600

① Alpha-Tastatur

Die Alpha-Tastatur des PC-1600 gleicht in ihrem Aufbau einer üblichen Schreibmaschinentastatur. Alle Tasten besitzen eine automatische Wiederholfunktion, d.h., werden sie für eine längere Zeit gedrückt gehalten, wird das ihnen zugeordnete Zeichen solange erneut geliefert, bis man diese wieder losläßt. Diese Wiederholfunktion läßt sich mit dem BASIC-Befehl KEYSTAT an- und abschalten.

② SML -Taste

Die SML-Taste (small key) kehrt das Verhalten der Alphatastatur um. Wird die SML-Taste betätigt, erscheint in der Statuszeile des Displays das Symbol SML. Der Computer befindet sich damit im Kleinschriftmodus, d.h., wird eine Aplhataste betätigt, erscheint grundsätzlich ein Kleinbuchstabe. Erst in Verbindung mit der SHIFT -Taste wird ein Großbuchstabe geliefert. Der Kleinschrift-Modus wird bei erneuter Betätigung der SML -Taste wieder verlassen.

③ SHIFT - Tasten

Zwecks bestmöglichen Bedienungskomforts ist der PC-1600 mit zwei SHIFT -Tasten ausgestattet. Beide sind in ihrer Funktion völlig identisch. Wie bei einer Schreibmaschine dienen sie der Umschaltung auf alternative Zeichen. Im Normalfall erhält man über die Tastatur den Zugriff auf die Zeichen, die auf den Tasten eingraviert sind. Betätigt man zuvor jedoch eine SHIFT -Taste, erhält man bei den Alphatasten die Kleinbuchstaben und bei allen anderen Tasten die Belegung, die über der Taste in oranger Farbe auf dem Gehäuse aufgedruckt ist. Im BASIC-Betrieb haben die SHIFT -Tasten noch weitere Funktionen (s. BASIC-Referenzteil).

④ Funktions-Tasten

Von den acht direkt unterhalb des Displays befindlichen Tasten, wirken sechs davon als spezielle Funktionstasten. Diese sind mit den Symbolen: ! " # $ % und & gekennzeichnet und werden im Verlaufe des Handbuches auch mit F1 bis F6 bezeichnet.

⑤ [ON] u. [OFF] Tasten

Mit diesen Tasten läßt sich der Computer ein- (ON) bzw. ausschalten (OFF). Bei eingeschaltetem Gerät wirkt die [ON] -Taste als BREAK-Taste, mit der sich laufende Programme abbrechen lassen.

⑥ Lösch-Taste

Die Taste [CL] löscht den Inhalt der Display-Zeile, in der sich der Cursor befindet. In Verbindung mit der [SHIFT] -Taste weist sie noch weitere Funktionen auf, die im BASIC-Referenzteil erklärt sind.

⑦ [MODE] -Taste

Diese Taste dient der Auswahl dreier wichtiger Betriebsarten. Es kann zwischen dem Programmier-Modus PRO, dem Programmablauf-Modus RUN und dem Funktionstasten-Belegungs-Modus RESERVE gewählt werden. Durch einfache Betätigung der [MODE] -Taste wird jedesmal eine Umschaltung zwischen PRO- und RUN-Modus vorgenommen. In den RESERVE-Modus gelangt man nur, wenn zuvor die [SHIFT] -Taste bedient wird. Durch eine einfache Betätigung der [MODE] -Taste wird der RESERVE-Modus wieder verlassen.

⑧ Numerische Tasten

Der abgesetzte Tastaturblock enthält Tasten, mit denen ein Zugriff auf Ziffern und arithmetische Operatoren besteht. Erst mit ihnen lassen sich numerische Ausdrücke programmieren oder aber über die Tastatur direkt auf manuelle Weise lösen. Wir nennen diese Tasten deshalb numerische Tasten. Der Gebrauch dieser Tasten ist im Kapitel 5 (Rechnen mit dem PC-1600) genauer beschrieben.

⑨ [ENTER] -Taste

Mit dieser Taste signalisieren Sie dem Computer, daß er die getätigten Eingaben als abgeschlossen ansehen und je nach Modus direkt verarbeiten oder als Programmzeile abspeichern soll. Sie hat zwar eine völlig andere Funktion, kann aber in gewisser Weise mit der Wagenrücklauftaste einer Schreibmaschine verglichen werden.

1.4 Anmerkungen zur Software

Ihr PC-1600 Taschencomputer basiert auf einem Z80-kompatiblen Mikroprozessor. Dieser Prozessor arbeitet nach Befehlen, die in einer sogenannten Maschinensprache abgefaßt sind. Alle Befehle, die Sie über die Tastatur in den Computer eingeben, sind Wörter der Programmiersprache BASIC. Diese Befehlswörter müssen intern in die entsprechenden Maschinensprache-Befehle umgesetzt werden, damit sie der Prozessor verstehen kann. Diese Arbeit nimmt Ihnen der eingebaute BASIC-Interpreter ab, der sich in einem Festwertspeicher, einem ROM-Baustein, befindet. Ihr persönlicher Dialog mit dem PC-1600 wird sich damit in der Regel über diesen Interpreter abspielen. Fortgeschrittene Programmierer haben aber auch die Möglichkeit, den PC-1600 in Maschinensprache zu betreiben. Hierfür ist eine Vielzahl spezieller Befehle (PEEK, POKE, CALL, BLOAD, BSAVE, usw.) bereitgestellt. Diese Befehle sollten aber nur dann angewendet werden, wenn tatsächlich Maschinensprache-Programme erstellt, geladen oder abgearbeitet werden sollen und eine genaue Kenntnis über die Speicher-Aufteilung besteht.

Das im PC-1600 implementierte BASIC stellt eine spezielle SHARP-Entwicklung für Taschencomputer dar und weist gegenüber anderen BASIC-Versionen einen erweiterten Befehlssatz auf.

Programme, die auf anderen BASIC-Computern entwickelt wurden, können durch sinnreiche Abänderungen in der Regel auch auf dem PC-1600 zum Laufen gebracht werden. Ein gesonderter Abschnitt dieses Handbuches geht auf die Kompatibilität zum PC-1500-BASIC ein und nennt notwendige Modifikationen.

Als Textverarbeitungssystem kann der PC-1600 in der vorliegenden Form nicht dienen. Die Befehle LOAD* und SAVE* ermöglichen aber eine eingeschränkte Textverwaltung. Es lassen sich damit kleine Textpassagen auf einem Datenträger speichern und wieder laden. Auf diesem Wege kann der PC-1600 auch als leistungsstarke Datenbank eingesetzt werden, wenn Sie ein geeignetes Verwaltungsprogramm dafür schreiben.

TEIL II

INBETRIEBNAHME UND GEBRAUCH DES PC-1600

KAPITEL 2 zeigt, wie die Batterien einzusetzen sind, wie der PC-1600 auf entladene Batterien oder Akkus aufmerksam macht und wie er über einen Netzadapter am Wechselstromnetz betrieben werden kann.

KAPITEL 3 beschäftigt sich mit dem praktischen Umgang des PC-1600. Sie lernen, wie man den Computer ein- und ausschaltet und wie man ihn im Falle eines Fehlers mit Hilfe eines RESETs wieder initialisieren kann.

KAPITEL 4 betrifft die Anzeige und zeigt, wie Sie anhand einer Statuszeile über die eingestellten Betriebsarten informiert werden und wie sich Datum und Uhrzeit der eingebauten Echtzeit-Uhr stellen lassen. Ebenfalls beschreibt dieses Kapitel die editierenden Tastenfunktionen.

KAPITEL 5 beschreibt abschließend, wie man den PC-1600 als Taschenrechner gebrauchen kann und einfache Rechnungen auf direktem Wege durchführt.

2. Stromversorgung des PC-1600

Ihr PC-1600 kann sowohl mit Batterien als auch mit Hilfe eines Netzadapters über das Wechselstromnetz betrieben werden. Zum Standardzubehör des PC-1600 zählen vier 1.5V-Batterien des Typs AA (bekannt als Mignonzelle).

Bei den mitgelieferten Batterien handelt es sich um normale Trockenbatterien, die NICHT wieder aufgeladen werden können !! Wiederaufladbare NiCad-Zellen sind wegen zu geringer Spannung nicht verwendbar !!

Die Lebensdauer der Batterien hängt davon ab, wie oft Sie den Computer benutzen und unter welchen Randbedingungen dieses geschieht. Typischerweise sollten die Batterien bei normalem Gebrauch (Verhältnis von Rechenzeit zu Anzeigezeit = 1:5) und einer Umgebungstemperatur von 25°C ca. 25 h halten.

2.1 Einsetzen der Batterien

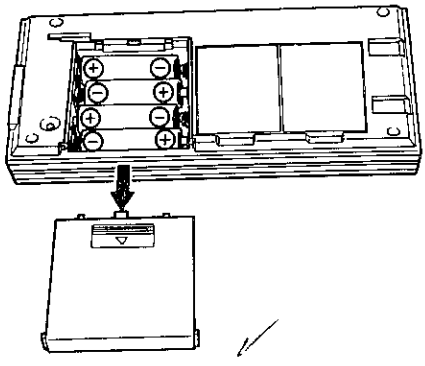


Abbildung 5 : Installation der Batterien

a) Entfernen Sie die Abdeckung des Batteriefaches, indem Sie diese in Pfeilrichtung herausziehen.

b) Setzen Sie die Batterien mit richtiger Polung, so wie in der Abbildung gezeigt, ein.

c) Schieben Sie die Abdeckung wieder über die Fachöffnung.

d) Befolgen Sie die Anweisungen des Abschnittes 3.3, der die Initialisierung (Reset) des Computers beschreibt.

Sind die mitgelieferten Batterien eines Tages verbraucht, müssen sie durch neue ersetzt werden. Beachten Sie dazu die im Anhang A enthaltenen Informationen bezüglich des Batteriewechsels.

2.2 Überwachung der Batterien

Verlieren die Batterien des Computers ihre Leistungskraft oder aber die Akkus des angeschlossenen Druckers CE-1600P (oder einer anderen Peripherie, die den PC-1600 mitversorgt), so erscheint in der Status-Zeile des Displays das Symbol BATT.

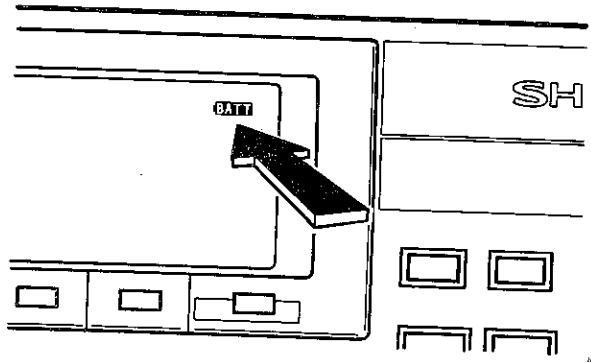


Abbildung 6 : Warnung vor zu schwachen Batterien

Mit Hilfe der Tasten [SHIFT] + [ON] läßt sich abfragen, ob die Batterien des Computers oder die Akkus der Peripherie diese Warnung verursachen. Bleibt nach Betätigung dieser Tastenkombination das Symbol weiterhin sichtbar, sind es die Batterien des Computers. Im anderen Falle sind die Akkus des Peripheriegerätes entladen. Sie sollten baldigst durch Anschluß eines Netzadapters wieder aufgeladen werden (s. Handbuch der Peripherie).

Verbrauchte Batterien sollten so bald wie möglich ausgewechselt werden !

Beachten Sie hierzu die in Anhang A beschriebenen Hinweise.

ACHTUNG:

Ein Batteriewechsel schließt immer das Risiko des Verlustes von Programmen und Daten ein. Beachten Sie deshalb bitte die genannten Hinweise sorgfältig, um diese Verluste zu vermeiden.

Ist ein Batteriewechsel zur Zeit nicht möglich, weil Sie z.B. noch ein wichtiges Programm sichern müssen, sollten Sie den PC-1600 vorübergehend über einen Netzadapter betreiben. Der Gebrauch des Netzadapters ist im folgenden Abschnitt erklärt.

Sofern Sie versuchen, den Computer bei Anzeige des BATT-Symboles trotzdem zu benutzen, stellt sich dieser automatisch aus.

Neben dem BATT-Symbol verfügt der PC-1600 noch über zwei weitere Warnungen vor zu geringer Stromversorgung. Diese tauchen aber nur dann auf, wenn der Computer an entsprechende Peripheriegeräte angeschlossen ist:

- Schaltet man den Computer ein, erscheint in der Anzeige die Meldung CHECK_ gefolgt von einem Nummern-Code für das betreffende Gerät.

- Während einer Programmdurchführung wird ein ERROR-Code ausgegeben, der Aufschluß über die Fehlerursache gibt.

Die CHECK-Meldungen sind in Abschnitt 3.1 (Einschaltung des PC-1600) erläutert. Anhang F gibt Auskunft über die Bedeutung ERROR-Codes.

Auf folgende Weise können Sie sich vergewissern, daß es sich bei der schlechten Stromversorgung nicht um die Batterien des Computers, sondern um die Akkus des Peripheriegerätes handelt:

Schalten Sie den Computer aus und wieder ein. Erscheint in der Anzeige eine CHECK-Meldung, sind die Batterien des Computers also in Ordnung. Stellen Sie anhand des Codes fest, welches Peripheriegerät die ungenügend geladenen Akkus aufweist und sorgen Sie für deren Wiederaufladung.

2.3 Anschluß des Netzadapters

Wird an die auf der Rückseite des PC-1600 befindliche Buchse ein Netzadapter angeschlossen, so kann der Computer aus dem Wechselstromnetz gespeist und damit die Batterien geschont werden.

Zwei Netzteiladapter sind je nach verwendetem Zubehör lieferbar:

- EA-160 (geliefert mit CE-1600P Drucker)
- EA-150 (geliefert mit CE-150 Drucker)

Netzadapteranschluß

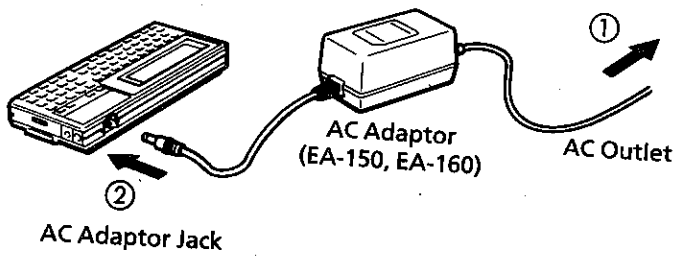


Abbildung 7 : Anschluß des Netzadapters

Beachten Sie beim Anschluß des Netzadapters folgende Schritte:

a) Schalten Sie den PC-1600 und jedwede Peripherie, die daran angeschlossen ist aus.

b) Stecken Sie zuerst den Netzstecker des Adapters in die Steckdose: (1).

c) Stecken Sie erst dann den Adapteranschluß in die dafür vorgesehene Buchse auf der Rückseite des PC-1600: (2).

d) Gehen Sie bei Entfernung des Adapters in umgekehrter Reihenfolge vor. Trennen Sie niemals den Adapter vom Computer solange dieser sich im eingeschalteten Zustand befindet.

e) Ziehen Sie den Netzstecker aus der Steckdose, wenn der Computer nicht in Gebrauch ist.

ACHTUNG:

Wenn im Computer keine Batterien eingesetzt oder diese unbrauchbar sind, hat die Entfernung des Netzadapters unweigerlich den Verlust von Programmen und Daten, die sich im Speicher des Computers befinden, zur Folge.

Der Netzadapter kann auch dazu verwendet werden, die im Drucker befindlichen NiCad-Akkus aufzuladen. Die Verwendung eines Netzadapters ist ein guter Weg, um die Daten vor einem Verlust beim Batteriewechsel zu bewahren (s. dazu Anhang A).

3. Ein- und Ausschaltung des Computers

Dieses Kapitel beschreibt die Methoden, wie sich der PC-1600 ein- und ausschalten läßt und zeigt Ihnen Beispiele, wie sich die unterschiedlichen Betriebsbedingungen auf die Anzeige nach dem Einschaltvorgang auswirken.

Der folgende Abschnitt betrifft den Gebrauch der RAM-Module und der Ein-/Ausgabe-Peripherie in Verbindung mit dem PC-1600.

Diese Module und Peripherie-Geräte sind ausführlich in Teil III behandelt. Anhang B widmet sich dem Austausch der Module. Für weiterführende Informationen sei auf die Bedienungsanleitungen verwiesen, die dem betreffenden Zubehör beiliegen.

3.1 Einschaltung des PC-1600

Der PC-1600 läßt sich auf drei verschiedene Arten einschalten:

- Manuell durch Betätigung der Taste [ON] .
- Automatisch durch Anwendung des BASIC-Befehles WAKE$, der den Computer zu einem vereinbarten Zeitpunkt, der durch die Angabe von Uhrzeit und Datum definiert ist, selbsttätig einschaltet.
- Ebenfalls automatisch durch Anwendung des WAKE$-Befehles, indem ein CI-Signal über das serielle RS-232C-Interface empfangen wird, das ein externes Gerät (z.B. ein Modem) liefert.

Was nach Einschaltung des Computers auf dem Display erscheint, ist abhängig davon, ob er mit optionalen RAM-Modulen bestückt oder an periphere Ein-/Ausgabegeräte angeschlossen ist. Die nachstehenden Beispiele mögen dieses verdeutlichen:

a) Manuelle Einschaltung des ohne Optionen versehenen PC-1600:

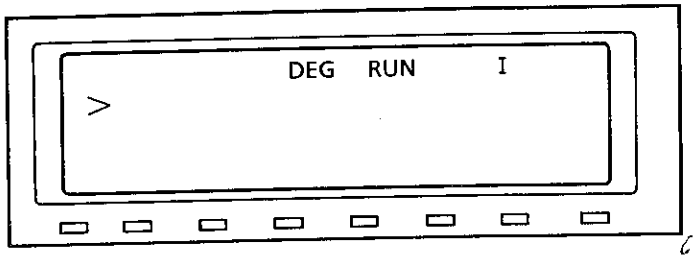


Abbildung 8

Was am anfang Erscheinen muß kann.

Am oberen Ende weist das Display eine Statuszeile auf, die Sie darüber informiert, welche Betriebsarten und Funktionen aktiv sind. Unmittelbar nach der ersten Einschaltung des Gerätes oder einer Betätigung der Reset-Taste sollte diese Statuszeile die folgenden Symbole anzeigen:

	DEG	für den Winkelmodus Altgrad (degrees)
	RUN	für den Betriebsmodus: Kommandoebene
und	I	für die erste RESERVE-Ebene.

Die Bedeutung dieser Symbole und der Modi werden im Verlaufe dieses Handbuches noch näher beschrieben.

Die Statuszeile erscheint während der gesamten Betriebsdauer des Computers. Da dieser im ausgeschalteten Zustand alle zuvor eingestellten Betriebsarten beibehält, sieht die Statuszeile nach dem Einschalten entsprechend aus und kann hier somit nicht allgemeingültig gezeigt werden.

Das Bereitschaftssymbol > Symbol (system prompt) zeigt an, daß der Computer darauf wartet, Eingaben entgegenzunehmen. Mit dem ersten eingegebenen Zeichen verschwindet dieses Symbol und es wird hinter das Zeichen der sogenannte CURSOR gesetzt. Dieser Cursor, der als Unterstrich dargestellt wird, verweist auf die Position, an der das nächste eingegebene Zeichen auf der Anzeige sichtbar werden wird.

b) Nach Einsetzung bzw. Austausch eines RAM-Modules

Bevor ein RAM-Modul gewechselt oder eingesetzt wird, muß der Computer unbedingt ausgeschaltet werden, um eine Zerstörung der Module oder des Computers zu vermeiden. Schalten Sie danach den Computer wieder ein, ergibt sich folgendes Bild der Anzeige:

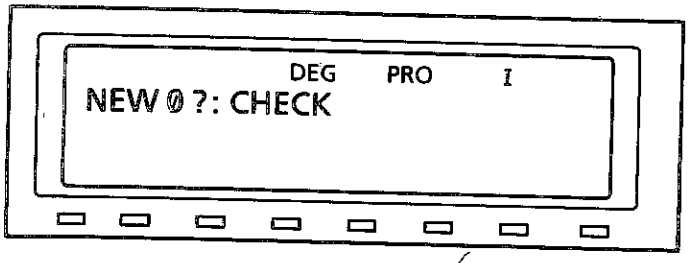


Abbildung 9

Diese Meldung teilt Ihnen mit, daß das System den zusätzlichen Speicherplatz entdeckt hat und darauf wartet, daß Sie die im Arbeitsspeicher möglicherweise vorhandenen Reste alter Daten löschen. Beachten Sie, daß diese Meldung nur dann erscheint, wenn die Module der Erweiterung des Arbeitsspeichers dienen.

Diese Meldung erscheint also nicht, wenn Sie Programm-Module einsetzen oder Module, die als RAM-Disk initialisiert worden sind. Die Anwendung der Module ist in Teil III dieser Dokumentation beschrieben.

Um den Arbeitsspeicher des Computers zu löschen, ist die Taste [CL] zu betätigen. Stellen Sie dann sicher, daß sich der Computer im PRO-Modus befindet. Drücken Sie hierzu gegebenenfalls die Taste [MODE]. Anschließend geben Sie nun den Befehl NEW0 ein und aktivieren ihn durch Bedienung der [ENTER] Taste.

In der Anzeige sollte dann, wie in der Abbildung gezeigt, das Bereitschaftszeichen > erscheinen. Nun können Sie den PC-1600 benutzen.

c) Einschaltung nach unsachgemäßer Einsetzung eines RAM-Modules

Der PC-1600 besitzt auf seiner Unterseite zwei Modulfächer: S1 und S2. In diese Fächer lassen sich unterschiedliche Module einsetzen, die entweder als Erweiterung des Arbeitsspeichers dienen oder aber zur Aufnahme von Programmen. Diese Module sind in Teil III dieses Handbuches ausführlich beschrieben.

Drei dieser zusätzlich erhältlichen Module, CE-151, CE-155 und CE-159, können ausschließlich in Fach S1 eingesetzt werden. Da beide Fächer in gleicher Form und Größe ausgeführt sind, kann es leicht vorkommen, daß sie versehentlich in das Fach S2 eingesetzt werden. Dort kann der Computer sie aber nicht nutzen. In einem derartigen Fall ergibt sich nachstehende Anzeige:

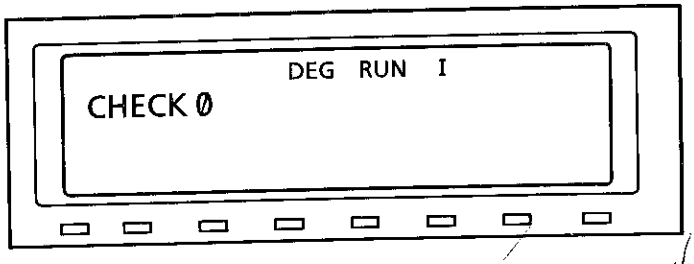


Abbildung 10

Diese CHECK-Meldung ist eine spezielle Maßnahme des PC-1600, um Sie darauf hinzuweisen, daß irgendetwas nicht in Ordnung ist. Prüfen Sie dann, welche Ursache für die Ausgabe dieser Meldung verantwortlich ist. Anhand der Nummer, die hinter dem Wort CHECK erscheint, läßt sich diese Ursache lokalisieren.

Die 0 in obiger Abbildung meint in diesem Falle ein Problem mit dem Modfulfach S2. Eine Tabelle mit den möglichen Codierungen wird im nächsten Abschnitt gegeben.

Sorgen Sie bei einer CHECK-Meldung umgehend für eine Abhilfe des Problemes, bevor Sie den Computer weiterbenutzen.

d) Wenn Peripherie-Geräte schwache Batterien aufweisen

Einige der zusätzlich erhältlichen Peripherie-Geräte, die an den PC-1600 angeschlossen werden können, besitzen eigene eingebaute Batterien bzw. Akkus. Sofern diese Energiequellen erschöpft und das betreffende Peripheriegerät am Computer angeschlossen ist, ergeben sich folgende Meldungen nach dem Einschaltvorgang:

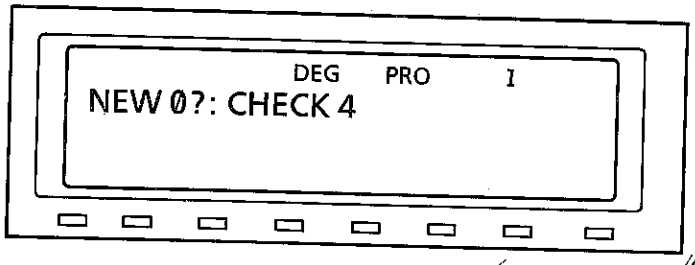


Abbildung 11

NEW0? sagt Ihnen, daß der Computer eine Änderung der zur Verfügung stehenden Speicherkapazität erkannt hat und mahnt Sie, den Arbeitsspeicher tunlichst zu löschen.

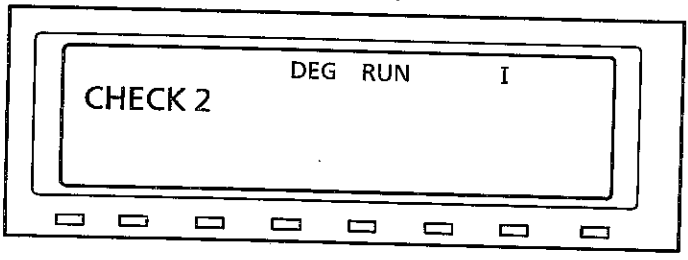


Abbildung 12

CHECK sagt Ihnen, daß die Akkus des angeschlossenen Gerätes erschöpft sind und einer Neuladung bedürfen.

Die nachfolgende Tabelle zeigt die Codes der CHECK-Meldungen: (Nicht in der Tabelle enthaltene Code-Nummern sind reserviert für zukünftiges Zubehör.)

CHECK Code	Fehlerort	Art des Fehlers
0	Modulfach S2	CE-151, CE-155 oder CE-159 ist fehlerhafterweise in diesem Fach eingesetzt worden. Setzen Sie Modul in das Fach S1 ein.
2	CE-1600F	Verbindung zum Drucker CE-1600P ist fehlerhaft. Die Akkus des Druckers sind nicht ausreichend geladen. Es liegt ein Defekt des Laufwerkes vor.
4	CE-1600P	Die Akkus des Druckers sind nicht ausreichend geladen. Es liegt ein Defekt des Druckers vor.
6	CE-150	Die eingebauten Akkus des CE-150 sind ungenügend geladen, oder es liegt ein Defekt dieses Druckers vor.
8	CE-158	Die Akkus dieser Seriell/Parallel Interface-Einheit sind nicht ausreichend geladen.

3.2 Ausschaltung des PC-1600

Der Computer PC-1600 kann sowohl manuell als auch automatisch ausgeschaltet werden. In jedem Falle bleiben dabei sämtliche im Speicher befindlichen Daten und Programme durch die Batteriepufferung (memory save guard) erhalten.

Die Ausschaltung kann gewollt hervorgerufen oder aber durch eine Selbstschutzmaßnahme bedingt sein. Im letzten Falle schaltet sich der PC-1600 selbsttätig ab, wenn folgende Bedingungen vorliegen:

- Wird innerhalb von ca. 10 Minuten keine Eingabe über die Tastatur des PC-1600 vorgenommen, so schaltet sich dieser selbsttätig ab, um die Batterien vor unnötiger Belastung zu schützen. Diese automatische Abschaltung, "auto power off" genannt, kann mit dem BASIC-Befehl POWER verhindert werden. Während eines laufenden Programmes findet diese selbsttätige Abschaltung nicht statt. Ein auf diese Weise ausgeschalteter Computer kann durch Betätigung der Taste [ON] wieder in Betrieb genommen werden. Alle zuvor in der Anzeige enthaltenen Daten werden dann wieder sichtbar.

- Versuchen Sie mit dem Computer zu arbeiten, obwohl in der Statuszeile mit dem BATT-Symbol auf eine unzureichende Stromversorgung hingewiesen wird, findet ebenfalls eine automatische Abschaltung statt. Sie sollten also in einem solchen Falle entweder die Batterien auswechseln oder das Gerät über einen Netzadapter betreiben.

Unter normalen Betriebsbedingungen stehen Ihnen zwei Wege zum Ausschalten des Computers offen:

- Manuell durch Betätigung der [OFF] Taste.
- Automatisch durch Anwendung des BASIC-Befehles POWER.

3.3 RESET-Möglichkeiten

Es gibt Fälle, in denen es angeraten oder notwendig erscheint, den Arbeitsspeicher des Computers zu löschen und systembedingte Voreinstellungen von Betriebsarten herzustellen. Einen solchen Vorgang bezeichnet man als Reset.

Der PC-1600 kennt zwei Reset-Arten: den einfachen Reset und den Total-Reset.

- Ein einfacher Reset wird durch Betätigung der auf der Rückseite des PC-1600 befindlichen RESET-Taste bewirkt.
- Ein Total-Reset erfolgt ebenfalls bei Betätigung dieser RESET-Taste, jedoch nur dann, wenn man gleichzeitig die Taste [ON] gedrückt hält.

Ein Reset ist unter folgenden Umständen erforderlich:

- Um den Computer zu initialisieren und nach dem Einsetzen bzw. dem Wechsel von Batterien für den normalen Betrieb vorzubereiten.
- Um wieder einen normalen Betriebszustand herzustellen, falls sich der Computer einmal "aufgehängt" haben sollte und z.B. wirre Zeichen auf dem Display ausgibt oder die Tastatur keine Reaktion zeigt usw.

Ein "Aufhängen" kann durch hohe Störimpulse, extreme Umgebungstemperaturen, unsachgemäße Programmierung und dergleichen mehr vorkommen.

Die Effekte, die in einem solchen Falle auftreten, können sehr variieren. Die Anzeige kann dabei "eingefroren" oder gelöscht werden. Tastendrücke erzeugen fehlerhafte oder unsinnige Zeichen in der Anzeige oder aber die Betätigung der Tasten bleibt völlig wirkungslos. Ein einfaches Ausschalten des Computers mittels der Taste [OFF] und eine Wiedereinschaltung mittels [ON] können in einem solchen Falle damit durchaus versagen.

Der auf der Rückseite des Computers befindliche RESET-Schalter dient der Behebung dieses Zustandes. Er ist tief im Gehäuse eingelassen, um eine versehentliche Bedienung auszuschließen.

WARNUNG:

Bei einem Reset können Daten und Programme verloren gehen!

Benutzen Sie zur Bedienung des RESET-Schalters einen spitzen Gegenstand, wie z.B. einen Kugelschreiber. Verwenden Sie aber auf keinen Fall Bleistifte oder Nadeln, d.h. Gegenstände, die abbbrechen könnten.

Sofern der Computer in die Drucker-Einheit CE-1600P eingesetzt ist, brauchen Sie ihn zum Zwecke eines Resets nicht hiervon lösen, da sich auf der Unterseite des Druckers ebenfalls ein RESET-Schalter mit gleicher Wirkung befindet.

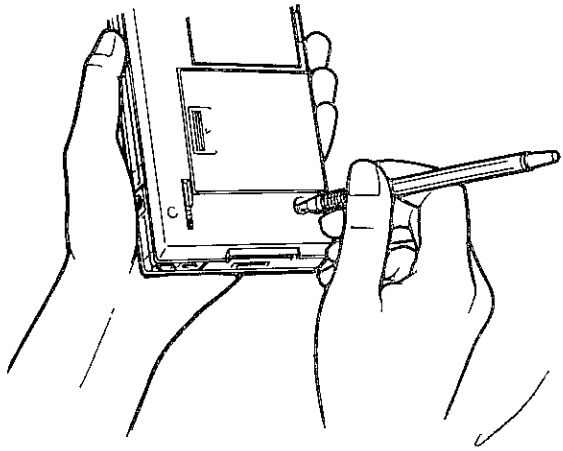


Abbildung 13 : Einfacher RESET

Sollte sich der Computer "aufgehängt" haben, probieren Sie, ihn zuerst mit einen einfachen Reset, der ähnlich wie ein Aus- und Einschalten wirkt, wieder zu einem normalen Betrieb zu bewegen. In den meisten Fällen wird dieses helfen und sowohl Programme und Daten erhalten bleiben, wenn auch nicht immer vollständig. Der Total-Reset ist radikal und sollte nur als letztes Mittel eingesetzt werden, wenn alle anderen Versuche scheitern.

Der Total-Reset wird auch dazu gebraucht, nach dem Wechsel oder dem ersten Einsetzen der Batterien eine Initialisierung des Computers vorzunehmen.

Die nachstehende Tabelle zeigt im Vergleich, welchen Einfluß die beiden RESET-Arten auf die sich ergebenden Einstellungen haben. Diese Einstellungen sind systembedingte Voreinstellungen, die man in englischer Sprache einfach als "defaults" bezeichnet.

Tabelle: Standard-Einstellungen nach Durchführung von Resets

	Total-Reset	Einfacher Reset
"auto power on" (WAKE$)	ausgeschaltet	unverändert
"auto power off"	10 Minuten	"
TITLE	SO:	"
Datum/Uhrzeit	01/01 00:00:00	"
Funktionstasten-Belegung	gelöscht	"
Kennwort ("password")	gelöscht	"
RS-232C-Interface (COM1)	1200,8,N,1,X,S	"
SIO-Interface (COM2)	38400,7,E,2,X,S	"
Maschinensprache-Bereich	0 Bytes	"
Tasten-Klick	ausgeschaltet	"
BEEP	eingeschaltet	"
Tasten-Wiederholfunktion	ausgeschaltet	"
Interrupt-Anweisungen	nicht erlaubt	"
Betriebsarten	RUN, DEG, I	können durch aufgetretenen Fehler verändert worden sein
BREAK ON/OFF	ON	unverändert
LOCK/UNLOCK	UNLOCK	"
MODE 0/1	MODE 0	"
MAXFILES	0	"
Speichererweiterungs-Modul	gelöscht	"
Programm-Modul	unverändert	"
RAM-Disk-Modul	unverändert	"
Interner Arbeitsspeicher	gelöscht	"

Durchführung eines einfachen RESETs

1) Drücken Sie den auf der Rückseite des Computers befindlichen RESET-Schalter für die Dauer einiger Sekunden.

2) Die Anzeige sollte dadurch gelöscht werden und in der linken oberen Display-Ecke das Bereitschaftszeichen > erscheinen, so wie nachfolgend illustriert:

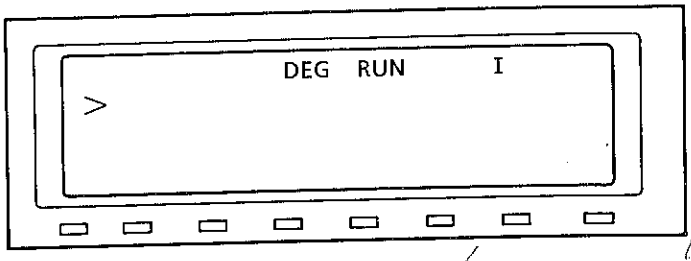


Abbildung 14

3) Dieses Zeichen ("prompt") bedeutet, daß der Computer bereit ist und auf Ihre Eingaben wartet. Ist nach dem Reset keine Löschung der Anzeige erfolgt, versuchen Sie abermals einen Reset auszulösen.

4) Nach Betätigung des RESET-Schalters erscheint manchmal das folgende Bild, sofern der Computer mit optionalem Zubehör betrieben wird:

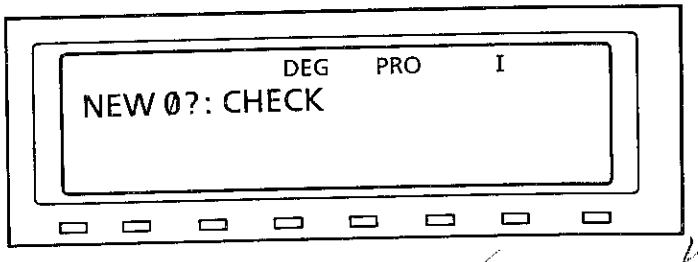


Abbildung 15

5) Um diese Anzeige zu löschen, drücken Sie bitte zuerst die Taste [CL] ("clear"). Vergewissern Sie sich dann, ob der PRO-Modus aktiviert ist. Betätigen Sie nötigenfalls dazu die Taste [MODE]. Geben Sie anschließend den Befehl NEW0 ein und drücken Sie die [ENTER] Taste.

Auf dem Display sollte dann das Bereitschaftszeichen > zu sehen und der Computer betriebsbereit sein, so wie es in der vorangegangenen Illustration gezeigt ist.

Durchführung eines TOTAL-RESETs

Folgen Sie den nachstehenden Schritten, wenn Sie die Batterien in den Computer zum ersten Male eingesetzt oder diese gewechselt haben.

In allen anderen Fällen lesen Sie bitte folgenden Hinweis, bevor Sie fortfahren:

WICHTIG: Sollte sich der Computer "aufgehängt" haben, so versuchen Sie ihn zuerst mit einem einfachen Reset in eine normale Betriebsart zu schalten, bevor Sie den weiteren hier beschriebenen Schritten folgen. Ein Total-Reset sollte nur dann ausgeübt werden, wenn Sie die damit verbundenen Folgen riskieren können. Beachten Sie hierzu die zuvor aufgeführte Tabelle, die die standardmäßigen Einstellungen nach einem solchen Reset beschreibt. Bedenken Sie, daß bei einem Total-Reset die in dem internen Arbeitsspeicher oder Modulen befindlichen Daten und Programme gelöscht werden.

1) Nehmen Sie den Computer so in die Hand, daß Sie leicht an die Taste [ON], die sich auf der rechten Seite der Tastatur befindet, herankommen. Während des eingeschalteten Zustandes wirkt diese Taste als BREAK-Taste.

2) Halten Sie die [ON] (BREAK)-Taste gedrückt während Sie den auf der Rückseite des Computers befindlichen RESET-Schalter betätigen.

3) Lassen Sie anschließend zuerst den RESET-Schalter los, bevor Sie danach auch die [ON] (BREAK)-Taste freigeben.

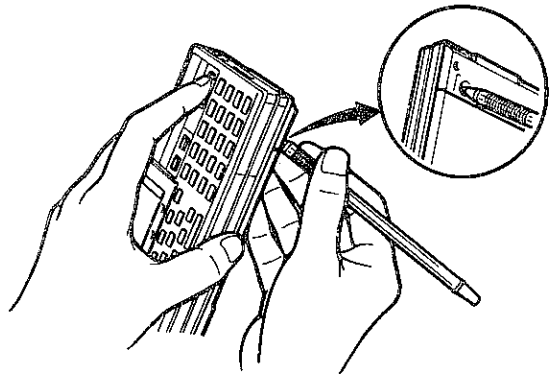


Abbildung 16 : TOTAL-RESET

4) Auf dem Display sollte dann, wie in der folgenden Abbildung gezeigt, das Bereitschaftszeichen > erscheinen.

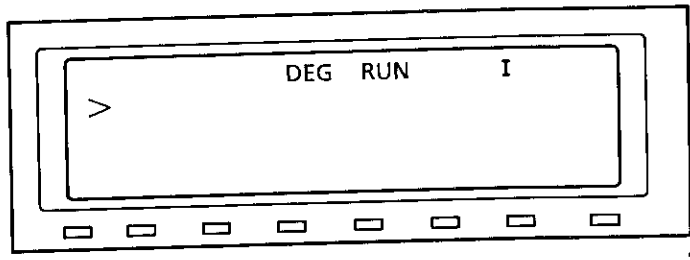


Abbildung 17

5) Das Bereitschaftszeichen bedeutet, daß der Computer bereit ist, Eingaben entgegenzunehmen. Sollte dieses Symbol nicht auf dem Display erscheinen, versuchen Sie einen neuen Total-Reset, so wie in den Schritten 1) bis 3) beschrieben.

WICHTIG:

Läßt sich der Computer weder durch einen einfachen noch einen totalen Reset in einen normalen Betriebszustand bringen, sollten Sie sich unverzüglich mit Ihrem SHARP-Händler in Verbindung setzen.

4. Das PC-1600-Display

Die Anzeige des PC-1600 besteht aus vier Zeilen, in denen sich jeweils 26 Zeichen darstellen lassen. Am oberen Display-Rand befindet sich eine Statuszeile, die Auskunft über die derzeitig eingestellten Betriebsarten und Funktionen gibt.

4.1 Anzeige-Modi

Es stehen zwei verschiedene Anzeige-Modi zur Verfügung, die über den BASIC-Befehl MODE angewählt werden können:

MODE 0 In dieser Betriebsart stehen in allen Fällen die vollen vier Display-Zeilen zur Verfügung. Sie können entweder jeweils bis zu 26 Zeichen darstellen oder aber als eine Einheit aus 156 * 32 Punkten angesprochen werden. Jeder Punkt ist dabei individuell setz- oder löschbar.

MODE 1 In dieser Betriebsart wird das einzeilige Display des Taschencomputers PC-1500 emuliert und damit die Daten ausschließlich in der untersten Display-Zeile angezeigt, wenn die Anweisungen PRINT und GPRINT zur Ausführung kommen. Allerdings besteht doch ein gewisser Unterschied zum richtigen PC-1500: Bei Eingaben werden zur besseren Übersicht auch die anderen drei Display-Zeilen benutzt. Nach welchen Regeln die Mitbenutzung der drei oberen Zeilen erfolgt, werden Sie beim Umgang mit dem PC-1600 kennenlernen.

4.2 Einstellung des Kontrastes

Sofern sich die umgebenden Lichtverhältnisse ändern oder aber der Betrachtungswinkel variiert wird, kann die Anzeige unter Umständen schlecht ablesbar sein. Deshalb ist der PC-1600 mit einem Kontrasteinsteller ausgerüstet, um jeweils die beste Ablesemöglichkeit zu gewährleisten. Dieser Kontrasteinsteller befindet sich auf der rechten Seite des Computers in der Nähe des seriellen RS-232C-Anschlusses.

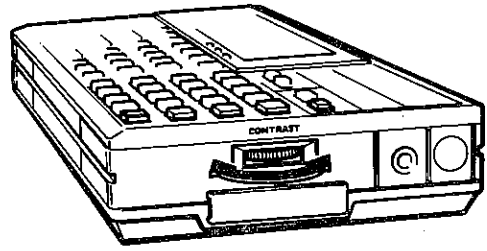


Abbildung 18 : Kontrasteinsteller

Drehen Sie diesen Einsteller so, daß die Anzeige wunschgemäß heller oder dunkler wird und eine einfache Ablesung ermöglicht. Sofern der Betrachtungswinkel oder die umgebenden Lichtverhältnisse geändert werden, können Sie mit diesem Einsteller den Anzeigekontrast erneut optimieren.

4.3 Symbole der Statuszeile

Am obersten Rand weist das LCD-Display eine Statuszeile auf, die mit speziellen Symbolen vor zu schwachen Batterien warnt sowie über die derzeitigen Betriebs-Arten informiert. Damit kann man mit einem Blick die Modus-Einstellungen überprüfen, also sehen, ob sie richtig gewählt worden sind.

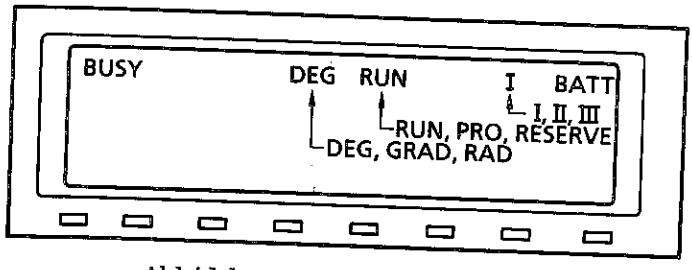


Abbildung 19 : Statuszeile

Nach Ausschaltung des Computers bleiben alle derzeitig gültigen Einstellungen erhalten, so daß nach erneutem Einschalten mit den zuvor gewählten Betriebsarten weitergearbeitet werden kann. Die Status-Zeile weist nach dem Einschalten des Computers immer den Zustand auf, der vor dem letzten Ausschalten herrschte.

Hier folgt nun eine Liste der Symbole, die in der Statuszeile erscheinen können und welche Bedeutung diese haben:

BUSY — Es läuft gerade ein Programm oder es wird ein Befehl abgearbeitet. Zu diesem Zeitpunkt können keine Eingaben über die Tastatur erfolgen.

SHIFT — Es ist soeben die [SHIFT] Taste betätigt worden. Diese Anzeige erlischt beim nächsten Tastendruck.

DEG — Winkel werden in Altgrad (degree) aufgefaßt.

RAD — Winkel werden im Bogenmaß (radian) aufgefaßt.

GRAD — Winkel werden in Neugrad (grad) aufgefaßt.

RUN — Der Computer befindet sich im RUN-Modus. Es können BASIC-Programme gestartet oder direkte Berechnungen über die Tastatur durchgeführt werden.

SMALL — Die Tastatur befindet sich im SMALL-Modus. In dieser Betriebsart liefern alle Alphatasten im Normalfall Kleinbuchstaben und in Verbindung mit der Umschalt-Taste [SHIFT] Großbuchstaben.

PRO — Der Computer befindet sich im PROgrammier-Modus. Es lassen sich Programme erstellen, anzeigen oder aber ändern.

RESERVE — Der Computer befindet sich im RESERVE-Modus. Somit können den programmierbaren Funktionstasten Zeichenketten (strings) zugeordnet werden.

DEF — Es wurde soeben die [DEF] Taste (define key) betätigt. Diese Anzeige erlischt beim nächsten Tastendruck.

I — Es ist die RESERVE-Ebene I aktiv.
II — Es ist die RESERVE-Ebene II aktiv.
III — Es ist die RESERVE-Ebene III aktiv.

CTRL — Die [CTRL] Taste (control key) wurde soeben betätigt. Mit dem nächsten Tastendruck erlischt diese Anzeige.

[BATT] — Die Batterien des PC-1600 oder die Akkus eines angeschlossenen Peripheriegerätes sind verbraucht.

[S] — Es ist die zweite Tastaturbelegung aktiv. Damit kann auf die internationalen Zeichen, die auf einer extra mitgelieferten Schablone vermerkt sind, zugegriffen werden.

4.4 Wahl der Betriebsarten

Der PC-1600 kennt drei grundlegende Betriebsarten, nämlich:

RUN, PRO und REVERSE.

Um entweder die Betriebsart RUN oder PROgram zu aktivieren, muß lediglich die [MODE] -Taste betätigt werden. Mit jedem Tastendruck wird von dem gerade herrschenden Modus in den jeweils anderen umgeschaltet. In den REVERSE-Modus gelangt man dagegen nur durch Betätigung der Tastenfolge [SHIFT] + [MODE] .

Je nach ausgewähltem Modus erscheint in der Status-Zeile des Displays ein entsprechendes Symbol. Der Gebrauch dieser drei Betriebs-Modi geht aus dem BASIC-Referenzteil hervor.

Hier nun eine Kurzbeschreibung dieser drei Modi:

RUN — Der RUN-Modus dient zur Durchführung direkter Berechnungen. Der PC-1600 wirkt dann wie ein Taschenrechner. Außerdem wird dieser Modus benötigt, um Programme zu starten, die im PRO-Modus erstellt worden oder aber von einem externen Speicher geladen worden sind.

PRO — Der PRO-Modus dient der Erstellung und der Auflistung sowie der Änderung von Programmen. In diesem Modus können jedoch keine BASIC-Programme gestartet werden.

RESERVE — Im RESERVE-Modus lassen sich die Funktionstasten mit Zeichenketten belegen, wobei jeweils drei Ebenen zur Verfügung stehen: I, II und III.

Unabhängig davon, welche Betriebsarten eingestellt worden sind, bleiben diese auch bei ausgeschaltetem Computer erhalten. Nach einer erneuten Einschaltung stehen sie dann wieder unverändert zur Verfügung. Hiervon können Sie sich anhand der Statuszeile überzeugen.

Auf diese Statuszeile sollten Sie auch immer dann einen Blick werfen, wenn es Probleme im Umgang mit dem PC-1600 gibt. Oftmals befindet er sich nur in einem falschen Betriebsmodus und verweigert daher mit Ausgabe eines ERROR-Codes die geforderte Tätigkeit. Anhang F beschreibt die Bedeutungen der ERROR-Codes.

4.5 Editierende Tastenfunktionen

Dieser Abschnitt beschreibt den Gebrauch der Editier-Tasten zur Bewegung des Cursors und zur Durchführung einfacher Korrekturen im RUN-Modus, wenn z.B. die Uhrzeit oder das Datum gesetzt oder direkte Rechnungen durchgeführt werden sollen. Weitergehende Editier-Funktionen, wie sie bei der Programmerstellung benötigt werden, sind in der Beschreibung des Editier-Modus erläutert, welcher im BASIC-Referenzteil zu finden ist.

Hier sind die Tasten und deren Funktionen, wie wir sie in diesem Kapitel verwenden wollen:

DEL
[◄] Cursor nach links bewegen

Dieses ist eine Taste von zweien, mit denen sich der Cursor innerhalb einer Zeile bewegen läßt. Bei ihrer Betätigung wird der Cursor um eine Zeichen-Position nach links versetzt. Hält man sie für längere Zeit gedrückt, wiederholt sich die Linksbewegung solange, bis die Taste wieder losgelassen wird oder aber der Cursor das logische Ende der Zeile auf der linken Display-Seite vorfindet.

SHIFT DEL
SHIFT [◄] Delete (Zeichen rechts vom Cursor löschen)

Bei Anwendung dieser Tastenkombination wird das Zeichen, welches vom Cursor berührt wird, gelöscht. Alle rechts vom Cursor befindlichen Zeichen wandern eine Position nach links. Die [SHIFT]-Taste ist nur für die Dauer einer [DEL] -Tastenbedienung wirksam. Somit muß [SHIFT] für jedes zu löschende Zeichen erneut betätigt werden. Bei eingeschalteter Tastenwiederholfunktion erübrigt sich dieser Umstand.

INS
[►] Cursor nach rechts bewegen

Bei Betätigung dieser Taste wird der Cursor um eine Position nach rechts versetzt. Wird diese Taste für längere Zeit gedrückt gehalten, so wiederholt sich diese Rechtsbewegung solange, bis man die Taste wieder losläßt oder der Cursor das logische Ende der Zeile vorfindet.

INS
SHIFT [►] Insert (Platzhalter einfügen)

Bei Bedienung dieser Tastenkombination wird in der Cursor-Position Platz für den Eintrag eines Zeichens geschaffen. Dazu verschieben sich alle Zeichen, die sich rechts von der Cursor-Position befinden sowie das vom Cursor verdeckte Zeichen selbst, gemeinsam um eine Position nach rechts. Da die [SHIFT] Taste nur für die Dauer eines nachfolgenden Tastendrucks wirksam ist, muß sie folglich für jeden Platzhalter, der eingefügt werden soll, erneut betätigt werden. Bei eingeschalteter Tastaturwiederholung läßt sich dieser Umstand umgehen.

[BS] Backspace (Zeichen links vom Cursor löschen)

Diese Taste bewegt den Cursor um eine Position nach links, wobei das dann vom Cursor verdeckte Zeichen gelöscht wird. Alle rechts vom Cursor befindlichen Zeichen, einschließlich desjenigen, das zuvor vom Cursor verdeckt gewesen ist, werden um eine Position nach links verschoben.

[CL] Clear (Löschung der Eingabezeile)

Ein Druck auf diese Taste löscht die gesamte Zeile, in der sich der Cursor befindet. Es erscheint dann das Bereitschaftszeichen am Anfang der Zeile und signalisiert damit, daß neue Eingaben gemacht werden können. Nach Eingabe des ersten Zeichens erscheint der Cursor.

[CTRL] [A] Insert/Overwrite (Editier-Modus-Umschaltung)

Durch Betätigung dieser Tastenkombination läßt sich zwischen dem Einfüge- (Insert) und dem Überschreibe-Mode (Overwrite) hin- und herschalten. Als Standard wird nach Einschaltung des Computers der Overwrite-Modus angenommen, bei dem bei jedem neu eingegebenen Zeichen an der jeweiligen Cursor-Position das alte Zeichen überschrieben wird. Im Insert-Modus werden dagegen alle rechts vom Cursor befindlichen Zeichen sowie das vom Cursor verdeckte Zeichen nach rechts verschoben und das neue Zeichen in die entstandene Lücke eingefügt.

Hier einige Punkte, die Sie sich merken sollten, um problemlos zu editieren:

- Die zweite Tastenbelegung, die jeweils oberhalb der Taste auf dem Gehäuse aufgedruckt ist, wirkt nur dann, wenn zuvor die Taste [SHIFT] betätigt worden ist. Falls Sie dieses versehentlich vergessen sollten, obwohl Sie gerade dabei sind eine Insert- oder Delete-Funktion auszuführen, wird in diesen Fällen lediglich der Cursor vorwärts oder rückwärts bewegt. Überprüfen Sie daher zuvor die Statuszeile, ob hier das Symbol für eine bereits betätigte [SHIFT]-Taste angezeigt wird.

- Die Betätigung der [SHIFT] -Taste ist nur für die Dauer des nächsten folgenden Tastendruckes wirksam. Um eine Reihe von Zeichen einzufügen oder zu löschen, weil beispielsweise ein Wort einzusetzen oder zu entfernen ist, muß diese Taste für jedes einzelne zu löschende oder einzusetzende Zeichen erneut betätigt werden. Beim Löschen sollte daher besser die Taste [BS] verwendet werden.

- Sobald Sie mit einer Eingabe beginnen, wandert der Cursor automatisch zur nächsten Position, bis Sie mit der Eingabe enden. Somit wird das gerade eingegebene Zeichen in die jeweils vom Cursor angezeigte Position geschrieben. Dieses gilt entsprechend, wenn Sie den Cursor vor einer Insert- oder Delete-Funktion verschieben.

- Für schnelle Korrekturen kurzer Eingaben führen Sie im Überschreibe-Modus (Overwrite) einfach den Cursor auf die nicht mehr benötigten Zeichen und ersetzen Sie diese durch die neuen. Ein Überschreiben löscht nur die Zeichen, die durch ein neues ersetzt werden. Es eröffnet weder Platz zur Einfügung weiterer Zeichen noch entfernt es den Zeilenrest.

- Der beste Weg sich mit den Editierfunktionen vertraut zu machen, sind praktische Übungen. Versuchen Sie es einfach. Falls Sie Probleme haben, schlagen Sie auf diesen Seiten nach, um sich zu erinnern.

4.6 Setzen von Uhrzeit und Datum

Die eingebaute Echtzeit-Uhr des PC-1600 erlaubt Ihnen, mit Hilfe geeigneter BASIC-Befehle den Computer zu einem definierten Zeitpunkt (bestimmt durch Datum und Tageszeit) automatisch Programme zu starten, den Computer einzuschalten, einen Alarmton von sich zu geben oder eine Meldung auf dem Display erscheinen zu lassen.

Dieser Abschnitt beschreibt, wie Sie Datum und Uhrzeit setzen können. Danach lassen sich die aktualisierten Werte jederzeit auf dem Display anzeigen oder von Programmen auswerten.

Die zeitbezogenen BASIC-Befehle lauten TIME, TIME$, DATE$, WAKE$ und ALARM$. Die ausführlichen Erklärungen hierzu finden Sie in Kapitel 14.

Den Befehl TIME werden wir in den folgenden Schritten dazu verwenden, um die Uhrzeit und das Datum zu initialisieren. Dieser Befehl kann auch zur Anzeige dieser Daten benutzt werden.

Erforderliche Schritte für den Setzvorgang:

1) Vergewissern Sie sich, daß sich der eingeschaltete Computer entweder im RUN oder PRO-Modus befindet. Stellen Sie dieses nötigenfalls durch Bedienung der [MODE] -Taste sicher.

2) Geben Sie hinter dem Bereitschaftszeichen > dann den Befehl TIME= ein, den Sie Zeichen für Zeichen über die Tastatur eintippen. Setzen Sie keine Leerstelle (space) zwischen die Buchstaben. Vergessen Sie ebenfalls nicht das Gleichheitszeichen. Sind diese Bedingungen erfüllt, ist der Computer bereit, Zahlenwerte für das Datum und die Uhrzeit entgegenzunehmen.

3) Das Format für die einzugebenden Werte schreibt jeweils zwei Zeichen für Monat, Tag, Stunde, Minute und Sekunden vor, die auch in genau dieser Reihenfolge einzutippen sind. Beachten Sie, daß zwischen den Stunden und Minuten ein Dezimalpunkt zu setzen ist.

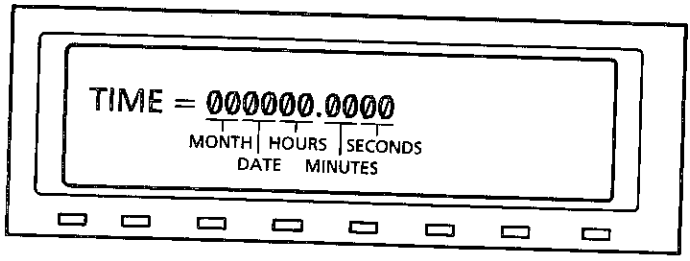


Abbildung 20 : Format der Zeitzuweisung

4) Bestehen die jeweiligen Werte nur aus einer einzelnen Ziffer (0 bis 9), so muß ihnen eine Null vorangestellt werden, um der Formatbedingung zu entsprechen. Vergessen Sie bitte nie den Dezimalpunkt zwischen der Stunden- und Minutenangabe. Unterbrechen Sie diese Zeitangaben nicht durch Leerstellen oder andere Zeichen. Benutzen Sie für die Uhrzeitangabe die 24-Stunden-Notation, wobei Mitternacht als 00 anzugeben ist.

5) Die nächste Abbildung zeigt ein Beispiel, bei dem das Datum auf den 24. Mai und die Uhrzeit auf 14 h 10 und 40 Sekunden gesetzt wird. Eine Jahreszahl ist nicht angebbar.

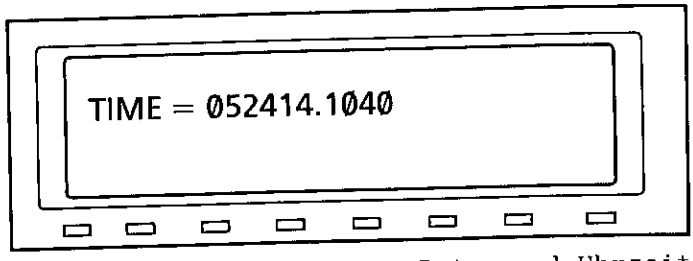


Abbildung 21 : Setzen von Datum und Uhrzeit

6) Tippen Sie nun einen Wert von 01 bis 12 für den Monat und einen Wert zwischen 01 bis 31 für den Tag ein. Fügen Sie an diese einen Wert von 00 bis 23 für die Stunden an und geben Sie dann einen Dezimalpunkt ein. Anschließend können Sie die Minuten und Sekunden jeweils mit Werten zwischen 00 und 59 eingeben.

7) Der Computer übernimmt die angegebene Zeit sekundengenau bei Betätigung der ENTER -Taste. Dieses ist sehr praktisch, wenn die Uhrzeit nach einer telefonischen Zeitansage gestellt werden soll.

8) Nun ist alles vorbereitet. Drücken Sie die ENTER Taste, um Datum und Uhrzeit zu setzen.

9) Immer dann, wenn Sie die aktuelle Uhrzeit wissen möchten, geben sie das Befehlswort TIME ohne Gleichheitszeichen ein und betätigen die ENTER Taste. Datum und Uhrzeit werden dann am Ende der nächsten Zeile angezeigt. Die nachstehende Abbildung zeigt, wie das Display in einem solchen Falle aussehen könnte.

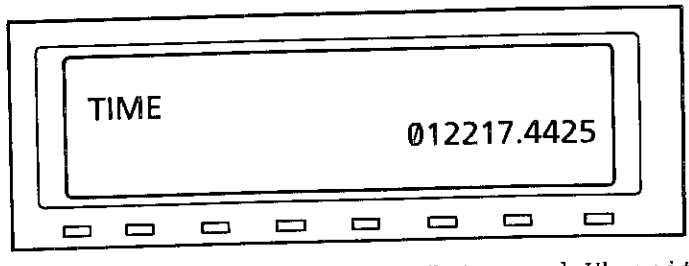


Abbildung 22 : Anzeige von Datum und Uhrzeit

5. Rechnen mit dem PC-1600

Dieses Kapitel zeigt Ihnen, wie sich mit dem PC-1600 eine Reihe von Berechnungen auf direktem Wege durchführen lassen, so als ob Sie einen Taschenrechner benutzten. Dazu wird Ihnen die separate numerische Tastatur eine wertvolle Hilfe sein, die eine schnelle Eingabe in gewohnter Weise erlaubt. Sofern der Computer dabei an einen der optionalen Drucker CE-1600P oder CE-150 angeschlossen ist, können alle eingegebenen Aufgaben und erhaltenen Ergebnisse mitprotokolliert werden. Auf diese Weise wirkt die Kombination aus Computer und Drucker ähnlich wie eine herkömmliche und mit einem Druckwerk ausgestattete Additionsmaschine. Natürlich kann der PC-1600 aber wesentlich mehr. Erst durch die Programmierung von Rechenausdrücken wird er Ihnen bei der Lösung von Problemen ein Hilfswerkzeug von besonders unschätzbarem Wert sein.

Falls Ihnen bei den Eingaben von Rechenausdrücken Fehler unterlaufen, schauen Sie bitte im Kapitel 4 nach, wie Sie diese mit Hilfe der Editier-Funktionen beseitigen können.

5.1 Wahl der Betriebsart

Für die Durchführung direkter Berechnungen sollte der Computer in den RUN-Modus geschaltet sein, obwohl sich fast alle Rechenfunktionen auch im PRO-Modus ausführen lassen. Dieser Modus ist jedoch dazu gedacht, wenn Sie später eigene Programme schreiben wollen, d.h. in BASIC programmieren möchten. Berechnungen, die mit einer Zahl beginnen, werden in diesem Modus irrtümlich als Eingabe von Programm-Zeilen interpretiert.

Vor Ausführung von direkten Berechnungen sollten Sie sich daher anhand der Status-Zeile des Displays vergewissern, ob sich der Computer im RUN-Modus befindet. Gegebenenfalls müssen Sie ihn erst durch die Betätigung der Taste [MODE] in diese Betriebsart schalten.

5.2 Tasten-Funktionen

Dieser Abschnitt führt in den Gebrauch des numerischen Tastenfeldes ein und beschreibt die speziellen Tastenfunktionen.

[ENTER] ENTER-Taste

Die [ENTER]-Taste signalisiert (wie Sie bereits aus den vorausgegangenen Abschnitten ersehen konnten) dem Computer, daß die Eingabe beendet ist und die Auswertung und Bearbeitung derselben erfolgen soll.

Bei einfachen Berechnungen entspricht diese Taste in ihrer Bedeutung dem Gleichheitszeichen in einer Rechenaufgabe. Benutzen Sie aber bitte nicht die Taste [=] , wenn eine Berechnung ausgelöst werden soll. Diese Taste ist ausschließlich für einen speziellen Gebrauch bei der BASIC-Programmierung reserviert.

[*] Multiplikations-Taste

Diese Taste liefert das Multiplikationszeichen, das in der Computer-Technik üblicherweise durch einen Stern dargestellt wird.

[/] Divisions-Taste

Der von links unten nach rechts oben verlaufende Schrägstrich (slash) dient in der Computer-Technik zur Darstellung des Divisions-Operators.

[E] Exponenten-Taste

Mit dem Buchstaben E kennzeichnet der Computer bei numerischen Werten, die er im wissenschaftlichen Format ausgibt, den Exponenten. Bei der Eingabe von Exponenten sind diese entsprechend zu kennzeichnen. Deshalb muß vor jeder Exponenteneingabe die Taste [E] betätigt werden.

5.3 Rechenbeispiele

Die nachfolgenden Beispiele sollen Ihnen nun zeigen, wie man mit dem PC-1600 direkte Berechnungen ausführt. Dazu ist jeweils die Aufgabenstellung in konventioneller Schreibweise dargestellt und darunter die Abbildung gezeigt, welche Tasten zur Eingabe sowie zur Lösung der Aufgabe zu betätigen sind. Anschließend ist das Resultat der Aufgabe genannt. Das letzte Beispiel zeigt, wie das eingebaute BASIC zur Lösung komplizierterer Aufgabenstellungen ausgenutzt werden kann.

Beispiel 1 : Aufgabe : $2 + 3 \times 4 = ?$

Eingabe : [2] [+] [3] [*] [4] [ENTER]

Resultat: 14

Beispiel 2 : Aufgabe : $36 : (1 + 2) = ?$

Eingabe : [3] [6] [/] [(] [1] [+] [2] [)] [ENTER]

Resultat: 12

Beispiel 3 : Aufgabe : $5 \times 10^3 : (4 \times 10^{-3}) = ?$

Eingabe : [5] [E] [3] [/] [4] [E] [-] [3] [ENTER]

Resultat: 1250000

Beispiel 4 : Aufgabe : $\sin^2 (30°) = ?$

Eingabe : [D] [E] [G] [.] [ENTER]

[(] [S] [I] [N] [3] [0] [)]

[SHIFT] [^] [2] [ENTER]

Resultat: 0.25

5.4 Kettenrechnungen

Wenn Sie die obigen Beispiele nachvollzogen haben, werden Sie bemerkt haben, daß das Resultat immer auf der rechten Seite der Zeile angezeigt wird, die der Zeile mit der Aufgabenstellung unmittelbar folgt.

Dieses Ergebnis kann als Grundlage weiterer Berechnungen dienen, ohne es erneut eintippen zu müssen. Die weiteren Operatoren und Zahlen brauchen dazu einfach nur an dieses Resultat angekettet zu werden. Sobald nach Anzeige des Resultates die Eingabe eines Operators erfolgt, kopiert der PC-1600 diesen Wert an den Anfang der neu begonnenen Zeile und setzt das Operatorzeichen dahinter. Sollte das Resultat in der letzten Zeile stehen und daher keine Zeile für die neue Rechnung zur Verfügung stehen, verschiebt der Computer den gesamten Display-Inhalt um eine Zeile nach oben und schafft somit in der letzten Display-Zeile Platz für die neuen Eingaben. Diesen Vorgang bezeichnet man als "Scrolling".

Versuchen Sie die in der Abbildung gezeigten Werte einzugeben und beobachten Sie, wie Ihr PC-1600 reagiert.

Eingabe:	Anzeige auf dem Display:
1 2 / 6 ENTER	12/6 2
* 5 + 1 0 ENTER	12/6 2 * 5 + 10 2 20
+ 3 0 ENTER	2 * 5 + 10 20 + 30 20 50

Abbildungen 23 -25

5.5 Rückholung eines Rechenausdrucks

In Kapitel 4 haben Sie gelernt, wie man den Cursor nach links und rechts bewegt und Eingaben editiert. Nun möchten wir Ihnen zwei weitere Funktionen dieser beiden Tasten nennen, die sich sehr angenehm bei direkten Berechnungen bemerkbar machen. Es sind hiermit die Rückruf-Funktionen (recall functions) dieser Tasten gemeint.

Mit diesen Funktionen läßt sich ein gerade berechneter Ausdruck erneut in die Anzeige holen und korrigieren. Diese Möglichkeit besteht in leicht abgewandelter Form auch dann, wenn auf dem Display ein ERROR-Code angezeigt wird, also vom Computer ein Fehler im Rechenausdruck bemerkt worden ist. Näheres zu diesen ERROR-Codes finden Sie im nächsten Abschnitt.

Ein Rechenausdruck kann nur dann zurückgeholt werden, wenn nach der Ausgabe seines Resultates keine weiteren Berechnungen oder Befehle durchgeführt worden sind. Bei erfüllter Bedingung kann mit den beiden Cursor-Tasten [►] und [◄] die Rückholung erfolgen.

Die Taste [►] setzt den als blinkenden Block dargestellten Cursor auf den Anfang der zurückgeholten Zeile. Die Taste [◄] setzt dagegen den Cursor hinter das Ende dieser Zeile, der dann durch einen nichtblinkenden Unterstrich dargestellt wird.

Nun kann mit diesen beiden Tasten der Cursor beliebig innerhalb des Ausdruckes hin- und hergefahren werden. Durch Überschreiben oder mit Hilfe der weiteren Editier-Tasten (s. Kapitel 4) lassen sich nun Änderungen vornehmen.

Dieses erweist sich als angenehm, wenn man testen möchte, wie sich Werteänderungen innerhalb einer Formel auf das Ergebnis auswirken, ohne jedesmal die Formel neu eintippen zu müssen. Ebenso hilfreich ist dieser Umstand, wenn das Ergebnis einem suspekt erscheint und eine fehlerhafte Werteingabe vorliegt.

Nachdem die Änderungen abgeschlossen sind, drücken Sie wieder die [ENTER] Taste, damit das Ergebnis des korrigierten Rechenausdruckes erscheint. Es wird unterhalb der Zeile dargestellt, in der sich zuvor der Cursor während des Editierens befand.

5.6 ERROR-Codes

Wird der Computer zur Berechnung eines Ausdrucks aufgefordert, so überprüft er zuerst, ob alle Eingaben zulässig sind. Falls er irgendetwas vorfindet, was ihm nicht korrekt erscheint, führt er die Berechnung nicht aus und zeigt stattdessen in der Zeile, in der sonst das Ergebnis angezeigt werden würde, einen ERROR-Code an. Ziehen Sie in einem solchen Fall die Tabelle der ERROR-Codes (siehe Anhang F) zu Rate, um die Bedeutung der Fehlermeldung zu erfahren und die Fehlerursache damit ausfindig machen zu können.

Ein typischer aber einfacher Fehler ist das Fehlen einer Klammer in komplizierteren Ausdrücken mit mehrern Klammerpaaren. Falls eine Fehlermeldung erscheint, muß diese zuvor quittiert, also gelöscht werden, um den Fehler beheben zu können.

Wählen Sie einen der folgenden Wege zur Fehlerbeseitigung:

- Benutzen Sie ähnlich wie bei den eben behandelten Rückruffunktionen die Tasten [►] und [◄] zur erneuten Anzeige des Rechenausdruckes. Mit Betätigung einer dieser Tasten wird die Fehlermeldung entfernt und an ihrer Stelle der kopierte Rechenausdruck angezeigt. Dabei befindet sich der als Block dargestellte und blinkende Cursor an der Position, wo der Computer einen Fehler festgestellt hat. Der fehlerhafte Ausdruck kann nun korrigiert und durch Druck auf die Taste [ENTER] erneut berechnet werden.

- Ist Ihnen die Fehlerursache jedoch ohne diese Hilfestellung bereits klar und möchten Sie den gesamten Rechenausdruck vollständig neu eingeben, so können Sie die Fehlermeldung mit folgenden Tasten quittieren, d.h. löschen:

  [ON] oder [CL]

  Mit der Tastenkombination [SHIFT] + [CL] ist ebenfalls eine Quittierung der Fehlermeldung möglich. Hierbei wird jedoch das Display gelöscht, so daß der fehlerhafte Rechenausdruck nicht als Grundlage einer korrigierten Eingabe abgelesen werden kann. Betätigen Sie nach Eingabe des korrigierten Ausdruckes in jedem Falle aber die [ENTER] Taste.

Die ERROR-Codes, die in diesem Abschnitt behandelt worden sind, stehen im Zusammenhang mit unzulässigen Angaben innerhalb von Rechenausdrücken. Erinnern Sie sich, daß Ihr PC-1600 auch andere ERROR-Codes ausgeben kann, die beispielsweise auf unbrauchbare Batterien usw. hinweisen. Ermitteln Sie anhand der in Anhang F aufgelisteten ERROR-Codes bitte immer erst die Bedeutung des angezeigten Codes, bevor Sie weitere Maßnahmen ergreifen.

5.7 "Arithmetische" BASIC-Funktionen

Ihr PC-1600 besitzt eine Vielzahl eingebauter Funktionen, die sowohl eine schnelle Berechnung auf direkte Art ermöglichen, als auch die indirekte Lösung komplexer Probleme mit Hilfe von Programmen. Zwei dieser Funktionen, SINus und DEGREE haben wir bereits kurz in unseren Rechenbeispielen aufgezeigt. Mit der nachfolgenden Liste sollen nun alle Rechen-Funktionen genannt werden, die das PC-1600-BASIC zur Verfügung stellt. Die genauen Beschreibungen dieser BASIC-Befehle können Sie im Kapitel 14 dieses Handbuches nachlesen. Beachten Sie hierzu aber auch die Abschnitte des BASIC-Referenzteiles (Teil IV), die den Gebrauch von Ausdrücken und Operatoren (Abschnitt 10.4) beschreiben. Dort finden Sie auch eine Übersicht, welche Priorität die einzelnen Operatoren haben, d.h., nach welcher Rangfolge sie vom Computer abgearbeitet werden.

Folgende Funktionen stellt das BASIC des PC-1600 bereit:

Funktion	mathematische Schreibweise	Bemerkung
ABS(X)	$\|x\|$	Absolutbetrag
ACS(X)	arccos(x)	Arcus-Cosinus
ASN(X)	arcsin(x)	Arcus-Sinus
ATN(X)	arctan(x)	Arcus-Tangens
COS(X)	cos(x)	Cosinus
DEG(X)		Winkelformat: Dezimal
DMS(X)		Winkelformat: Stunde,Min.,Sek.
EXP(X)	e^x	Exponential-Funktion
INT(X)	int(x)	Integer-Funktion, Ganzzahlwert
LN(X)	ln(x)	Natürlicher Logarithmus
LOG(X)	lg(x)	Dekadischer Logarithmus
PI	π	Wert der Zahl
RND(N)		Erzeugt Zufallszahl
SGN(X)	sgn(x)	Vorzeichen (Signum)
SIN(X)	sin(x)	Sinus
SQR(X)	$\sqrt{x}$	Quadratwurzel
TAN(X)	tan(x)	Tangens

TEIL III

BENUTZUNG DER SCHNITTSTELLEN UND OPTIONEN

Ein wesentlicher Schlüssel zur Flexibilität Ihres PC-1600-Taschencomputers liegt darin begründet, ihn zu einem kompakten System mit zusätzlichen Ein-/Ausgabe-Geräten und anderen Optionen aufrüsten zu können.

KAPITEL 6 gibt einen System-Überblick und zeigt die Erweiterungsmöglichkeiten durch RAM-Module, Drucker, Cassetten-Recorder und Diskettenlaufwerk auf. Außerdem beschreibt dieses Kapitel die seriellen Schnittstellen und den Analog-Eingang.

Kapitel 7 nennt dasjenige PC-1500-Zubehör, das auch in Verbindung mit dem PC-1600 verwendet werden kann.

6. Erweiterung des PC-1600

6.1 System-Überblick

Trotz seiner geringen Ausmaße enthält der PC-1600 eine gewaltige Anzahl leistungsstarker Funktionen. Noch leistungsstärker und vielseitiger wird der SHARP PC-1600 jedoch, wenn man ihn an die speziell für ihn konstruierten Peripheriegeräte anschließt oder ihn mit anderen speziellen Optionen betreibt. Ebensogut kann er auch mit den Optionen des PC-1500 betrieben werden. Wie auch immer, mit dem PC-1600 als Kern läßt sich ein kompaktes System aufbauen, das den verschiedensten Bedürfnissen im Bereich der Datenverarbeitung, -speicherung und -ausgabe gerecht wird.

Dieser Abschnitt des Handbuches gibt einen kurzen Abriß über die lieferbaren Optionen und erwähnt die markanten Punkte, die es im Umgang mit ihnen zu beachten gilt. Für weitergehende Erklärungen verweisen wir auf die jeweiligen Bedienungsanleitungen, die den Optionen beigefügt sind.

Der Computer weist drei Schnittstellen auf: ein SIO-Interface (optoelektronische Schnittstelle), ein RS-232C-Interface und einen Analog-Eingang. Dadurch können an den Computer externe Geräte, wie Drucker, Modem, Barcode-Leser, Sensoren usw. direkt angeschlossen werden.

Mit zusätzlichen RAM-Modulen, die man in die dafür vorgesehenen Fächer auf der Unterseite des PC-1600 einsetzen kann, läßt sich der 16 KByte umfassende interne Arbeitsspeicher auf bis zu ca. 80 KByte erweitern. Diese Module können auch als Programm-Module initialisiert werden und der Speicherung von Programmen dienen, die man jederzeit schnell verfügbar haben möchte.

Der Computer PC-1600 läßt sich direkt in den Drucker CE-1600P einsetzen, mit dessen Hilfe Sie vierfarbige Texte oder Grafiken zu Papier bringen können. Diese kompakte Einheit aus Drucker und Computer bildet die Basis für den Anschluß weiterer Optionen. So kann als nützlicher Zusatz das Disketten-Laufwerk CE-1600F mit in diese Einheit integriert werden und der Massenspeicherung von Daten auf Disketten dienen.

Ein weiteres herausragendes Merkmal des PC-1600 ist seine volle Kompatibilität zu seinem Vorgänger. Damit kann er nicht nur die Befehle und Programme des PC-1500 verarbeiten, sondern auch mit vielen PC-1500-Optionen betrieben werden. Dieses beinhaltet auch den Gebrauch des Cassetten-Recorders, der RAM-Module, der Ein- und Ausgabe-Geräte sowie der parallelen und seriellen Ports.

Die nachfolgend gegebene Abbildung zeigt, in welcher Vielfalt sich aus dem PC-1600-Taschencomputer mit den unterschiedlichsten Optionen ein individuelles System gestalten läßt. Dabei sind die kompatiblen PC-1500-Optionen schattiert dargestellt. In Kapitel 7 finden Sie eine Zusammenstellung, welche PC-1500-Optionen mit dem PC-1600 kombiniert werden können.

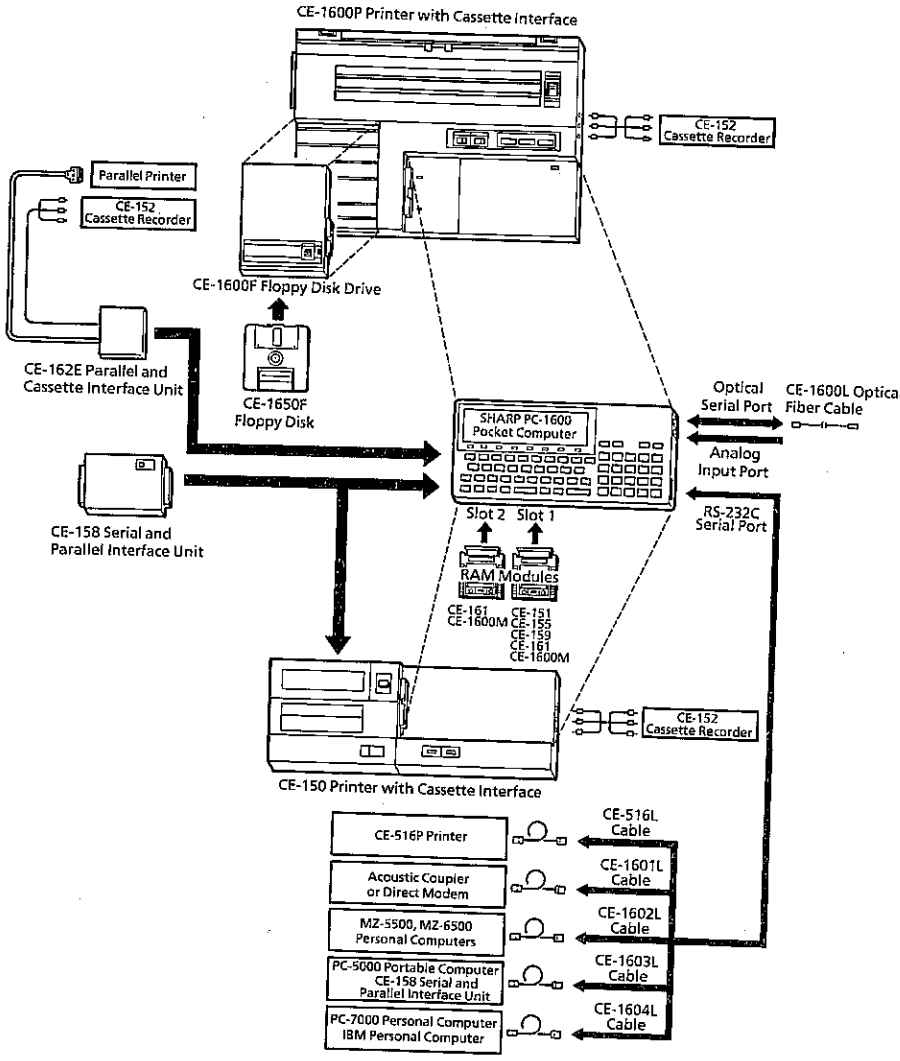


Abbildung 26 : System-Überblick

6.2 Erweiterung durch RAM-Module

Den 16 KByte umfassenden internen Arbeitsspeicher können Sie mit Hilfe von RAM-Modulen auf ca. 80 KByte erweitern. Dazu brauchen die Module lediglich in die dafür vorgesehenen Fächer S1 oder S2 eingesetzt zu werden, die sich auf der Unterseite des Computers befinden. Dieser Abschnitt beschreibt, wie die unterschiedlichen Modul-Typen gebraucht werden und gibt einige Hinweise zu deren Installation sowie den mit Ihnen in Verbindung stehenden BASIC-Befehlen und Fehlermeldungen.

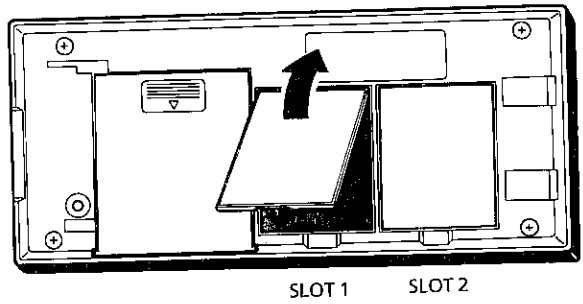


Abbildung 27 : Modul-Fächer

RAM-Module werden in zwei Versionen geliefert: Speicher-Module (memory modules) zur Erweiterung des internen Arbeitsspeichers und Programm-Module, welche eine interne Pufferbatterie zum Erhalt der Daten besitzen sowie mit einem Schreibschutzschalter ausgerüstet sind, um die Daten und Programme vor versehentlichem Beschreiben zu schützen. Programm-Module erlauben Ihnen hierin Programme hineinzuschreiben und auf lange Sicht zu speichern, so daß die betreffenden Programme einfach durch Einsatz der Module bei Bedarf schnell und bequem in den Computer geladen werden können.

Eine Besonderheit der Module CE-161 und CE-1600M ist es, daß sie wie Disketten formatiert und beschrieben werden können und damit als sogenannte RAM-Disks fungieren.

Nachstehende Tabelle zeigt einen Vergleich der für den PC-1600 lieferbaren RAM-Module:

a) Tabelle der RAM-Module

Modul-Bezeichnung	CE-151	CE-155	CE-159	CE-161	CE-1600M
Modul-Typ					
Speicher-Modul	x	x			
Programm-Modul			x	x	x
Kapazität	4 KB	8 KB	8 KB	16 KB	32 KB
Nutzbar als					
Speichererweiterung	x	x	x	x	x
Programm-Speicher			x	x	x
RAM-Disk				x	x
Batteriepufferung			x	x	x
Schreibschutz			x	x	x
Einsetzbar in Fach	1	1	1	1/2	1/2

b) Hinweise zur Benutzung der Module

1) Ausschaltung

Schalten Sie grundsätzlich den Computer aus, bevor Sie ein RAM-Modul einsetzen bzw. austauschen. Ist der Computer ausgeschaltet, sorgt ein Speicherschutz für den Erhalt seiner in ihm befindlichen Daten. Dieses gilt sowohl für die Module als auch den Arbeitsspeicher des Computers.

2) Einschaltung

Entfernen Sie niemals ein RAM-Modul, wenn der Computer eingeschaltet ist. Schalten Sie den Computer nach dem Einsatz oder Wechsel eines Modules ein, fordert er Sie durch eine Meldung dazu auf, den veränderten Speicherbereich zu löschen und zu initialisieren. Diese Aufforderung unterbleibt, wenn Sie ein Programm-Modul einsetzen oder ein Modul, das als sogenannte RAM-Disk initialisiert worden ist. Beachten Sie hierzu auch den Abschnitt 3.1 (Einschaltung des PC-1600).

3) Modul-Fächer

Auf der Unterseite des Computers befinden sich zwei Fächer, die der Aufnahme von Modulen dienen. Vergewissern Sie sich anhand der vorangegangenen Tabelle, welche Module nur in das Fach S1 eingesetzt werden dürfen. Lesen Sie bitte auch den Abschnitt 3.1, damit Sie die Bedeutung einer CHECK-Meldung verstehen.

4) Handhabung der Module

Berühren Sie niemals die Modul-Kontakte oder die Anschlüsse in den Modul-Fächern. Statische Entladungen könnten sowohl die Module als auch den Computer elektrisch zerstören. In mit Teppich ausgelegten Räumen sollten Sie, besonders wenn ein trockenes Winterwetter herrscht, unbedingt für die Entladung der Ihnen womöglich anhaftenden elektrischen Ladung sorgen, bevor Sie mit den Modulen hantieren. Die Entladung können Sie durch Berührung einer Türklinke oder anderer metallischer Objekte erreichen.

5) Schreibschutz-Schalter

Programm-Module haben einen Schreibschutz-Schalter, mit dem sich die im Modul gespeicherten Daten und Programme gegen ein versehentliches Überschreiben schützen lassen. Dazu muß sich der Schalter in Stellung ON befinden. Bei der beabsichtigten Programmierung der Module muß dieser in Stellung OFF gebracht werden.

6) Batterie-Pufferung

Alle Programm-Module enthalten eine Lithium-Zelle, um so die gespeicherten Daten auch dann zu erhalten (zu puffern), wenn sich die Module außerhalb des Computers befinden. Module ohne eine solche Batterie, also Speicher-Module, verlieren ihren Inhalt bei Entnahme aus dem Computer.

7) INIT-Befehl

RAM-Module können zu dreierlei Zwecken eingesetzt werden: als Speicher-Module zur Erweiterung des Arbeitsspeichers oder als Programm-Module zur Aufnahme und Speicherung von Programmen oder als RAM-Disketten. Benutzen Sie den BASIC-Befehl INIT, um die gewünschte Einsatzart zu initialisieren.

8) Referenz-Hinweise

- Halten Sie sich an die Bedienungshandbücher der Module, um Informationen zu erhalten über:

  - ihren Gebrauch,
  - zu beachtende Vorsichtsmaßnahmen,
  - technische Spezifikationen,
  - Austausch der Modul-Batterien.

- Ziehen Sie das hier vorliegende Handbuch zu Rate, wenn Sie Informationen benötigen über:

  - die zugehörigen BASIC-Befehle,
  - die Ein- und Ausschalt-Bedingungen,
  - die Modulfächer,
  - Hinweise zum Austausch.

c) RAM-Disk-Befehle

Die folgenden Befehle dienen dem Gebrauch eines Programm-Modules als RAM-Disk. Diese Befehle entsprechen genau jenen, die auch beim Umgang mit dem Diskettenlaufwerk Verwendung finden.

BLOAD — Lädt ein Maschinen-Programm von der RAM-Disk in den Arbeitsspeicher.

BSAVE — Sichert ein im Arbeitsspeicher befindliches Maschinensprache-Programm auf der RAM-Disk.

CLOSE — Schließt eine RAM-Disk-Datei und beendet damit die Zugriffsmöglichkeit auf diese.

COPY — Kopiert eine Datei von oder auf die RAM-Disk bzw. von externen Ein-/Ausgabe-Geräten.

EOF — Liefert einen Wert, der das Ende einer sequentiell eingelesenen Datei kennzeichnet.

FILES — Zeigt das Inhaltsverzeichnis der näher spezifizierten Dateien an.

INIT — Initialisiert und spezifiziert den Gebrauch der RAM-Module.

INPUT#	Liest einen Datensatz von einer sequentiellen Datei.
KILL	Löscht eine Datei von der RAM-Disk.
LFILES	Listet ein Inhaltsverzeichnis der Dateien, die sich auf der RAM-Disk befinden, auf dem Drucker CE-1600P auf oder sendet es über ein Interface an ein externes Gerät.
LOAD	Lädt eine Datei von der RAM-Disk.
LOAD*	Lädt eine aus Kommentarzeilen bestehende Datei von der RAM-Disk in den Arbeitsspeicher.
LOC	Liefert die Nummer der gelesenen bzw. geschriebenen Datensätze (records) einer geöffneten Datei.
LOF	Liefert die Größe einer Datei (angegeben in Bytes).
MAXFILES	Bestimmt, wieviele Dateien gleichzeitig geöffnet sein dürfen.
NAME	Ändert den Namen einer Datei.
OPEN	Öffnet eine RAM-Disk-Datei und erlaubt somit den Lese- bzw. Schreibzugriff auf diese.
PRINT#	Schreibt Daten sequentiell in eine RAM-Disk-Datei.
SAVE	Sichern eine Programm als Datei auf der RAM-Diskette.
SAVE*	Sichert Kommentarzeilen als Datei auf einer RAM-Disk.
SET	Setzt oder löscht den Schutz von RAM-Disk-Dateien.
TITLE	Selektiert ein in Modulfach S1 oder S2 befindliches Modul zur Programm-Eingabe oder Ausführung. Dieser Befehl ist nicht zulässig, wenn das Modul als RAM-Disk initialisiert ist.

d) ERROR-Codes bei Verwendung von Modulen

Die folgende Liste bezieht sich auf die Fehlermeldungen, die direkt im Zusammenhang mit dem Gebrauch von Modulen stehen. Beachten Sie auch die ERROR-Codes für die Datei- und Disketten-Fehler, die im Abschnitt über das Diskettenlaufwerk in Anhang F dieses Handbuches aufgeführt sind.

Code	Beschreibung
27	Unzulässiger Befehl. Adressierte Peripherie ist nicht angeschlossen.
101	Ungültiger Gerätename bei Verwendung der Befehle TITLE und NEW.
102	Ungültige Geräteauswahl (Gerät nicht angeschlossen)
103	Speicher des RAM-Modules ist voll. INIT-Parameter können nicht gesetzt werden.

6.3 Serielle Schnittstellen

Der Computer PC-1600 verfügt über zwei serielle Schnittstellen: Ein RS-232C-Interface und eine optoelektronische Schnittstelle, die mit SIO gekennzeichnet ist und in dieser Dokumentation oft auch als SIO-Interface bezeichnet wird. Die folgenden Abschnitte beschreiben den Gebrauch dieser Schnittstellen, listen deren Spezifikationen auf, zeigen deren Anschlußbelegungen und geben einen Überblick über die mit ihnen verknüpften BASIC-Befehle und ERROR-Codes.

A) RS-232C-Schnittstelle

Das RS-232C-Interface genügt den Anforderungen des EIA-Standards für asynchrone Datenübertragung zwischem dem PC-1600-Computer und anderen Geräten, wie z.B. einem seriellen Drucker, einem Modem oder einem anderen Computer.

In BASIC-Befehlen wird das RS-232C-Interface durch den Parameter COM1: angesprochen und kann nicht gleichzeitig mit der optischen Schnittstelle COM2: betrieben werden.

Das RS-232C-Interface des PC-1600 unterstützt ein CI-Signal, das z.B. von einem Modem geliefert wird, um den Computer automatisch einschalten zu können oder eine Interrupt-Routine ausführen zu lassen.

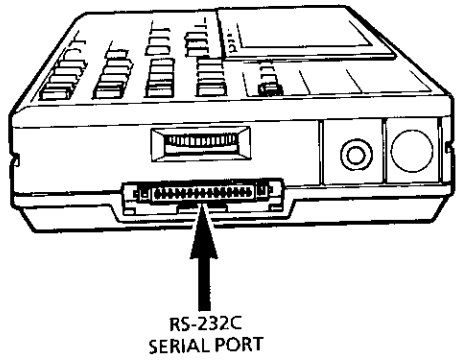


Abbildung 28 : Serieller Port

a) Hinweise zur Benutzung des RS-232C-Ports

1) Ausschaltung

Schalten Sie den Computer aus, wenn sie ihn an ein externes Gerät anschließen oder davon entfernen.

2) Handhabung

Berühren Sie niemals die Anschlüsse am Computer an den Kabeln oder an den Peripheriegeräten. Schützen Sie die Kontakte mit den dafür vorgesehenen Abdeckungen, solange sie nicht genutzt werden.

3) Übertragungsprotokoll

Die zwischen einem Computer und anderen Geräten stattfindende Datenübertragung bedarf eines Datenübertragungsprotokolles, damit die Daten ordnungsgemäß ausgetauscht und auf eventuelle Übertragungsfehler hin überprüft werden können. Solch ein Datenaustausch funktioniert allerdings nur dann problemlos, wenn die kommunizierenden Geräte mit demselben Übertragungsprotokoll arbeiten. Um die Parameter eines solchen Protokolles einstellen zu können, sind spezielle BASIC-Befehle vorgesehen, die in Kapitel 14 näher erläutert sind.

4) Geräteanschluß

Stellen Sie sicher, daß das mit dem Computer kommunizieren sollende Gerät an den richtigen Port angeschlossen ist, bevor Sie die Datenübertragung starten. Werden Daten gesendet, ohne daß ein Empfänger angeschlossen ist, hat dieses zur Folge, daß sich der Computer "aufhängt". Drücken Sie dann die Taste [ON] (BREAK), um einen Reset ohne Dateiverlust durchzuführen.

5) Referenz-Hinweise

- o Benutzen Sie die Bedienungsanleitungen der Kabel sowie der angeschlossenen Geräte (wie Drucker, Modem, Computer), um sich zu informieren über:

  - die Kommunikationsparameter,
  - die Kabelspezifikationen,
  - die Anschlußbelegungen,
  - die Sende- und Empfangs-Prozeduren.

- o Schauen Sie in das hier vorliegende Handbuch, um folgende Informationen zu erhalten:

  - die zugehörigen BASIC-Befehle,
  - das PC-1600-Kommunikationsprotokoll
  - die Protokoll-Parameter
  - die RS-232C-Anschlußbelegung
  - die ERROR-Codes.

b) Standardmäßige RS-232C-Einstellungen

Die nachfolgende Liste zeigt die standardmäßigen bzw. voreingestellten Kommunikations-Parameter für den RS-232C-Port (COM1:). Diese Einstellungen lassen sich mit den beiden BASIC-Befehlen SETCOM und PCONSOLE ändern.

Befehl	Parameter	Standard-Einstellung
SETCOM	Übertragungsrate	1200 baud (Bit/s)
	Wortlänge (5 bis 8 Bits)	8 Bits
	Parität ((un)gerade, keine)	keine
	Stop-Bit-Anzahl (1 oder 2)	2
	X-ON/OFF-Protokoll	ja
	SHIFT-IN/OUT-Protokoll (nur für 7 Bits)	ja
PCONSOLE	Zeilenlänge	unbegrenzt
	Zeilenende-code (EOL-Code)	<CR> + <LF>

c) RS-232C-Anschlußbelegung

Die nachstehende Tabelle zeigt, mit welchen Signalen die Pins belegt sind und wie diese Signale symbolisiert werden:

Pin	Signal	Symbol(e)	Kurzbeschreibung
2	Send Data (Ausgang)	TXD (SD)	Sendet serielle Daten vom Computer nach außen.
3	Receive Data (Eingang)	RXD (RD)	Empfängt Daten, die von außen geliefert werden.
4	Request to Send (Ausgang)	RTS (RS)	Aufforderung, Daten an den Computer zu senden.
5	Clear to Send (Eingang)	CTS (CS)	Externes Gerät möchte Daten senden.
6	Data Set Ready (Eingang)	DSR (DR)	Externes Gerät ist bereit für Transfer.
7	Signal Ground	SG	Signal-Masse.
8	Carrier Detect (Eingang)	CD	Eingang für den Signalträger.
9	Calling Indicator (Eingang)	CI	Interrupt-Anforderung vom externen Gerät.
10	Logic Voltage (Ausgang)	Vcc	Versorgungsspannung der Logik (4 bis 4.7 V).
14	Data Terminal Ready (Ausgang)	DTR (ER)	Computer ist bereit für Datentransfer.

Anmerkungen

1) Die Ausgangssignale RTS und DTR können auf "high" oder "low" gesetzt werden durch den BASIC-Befehl OUTSTAT.

2) Die Pegel der Eingangssignale CTS, DSR, CD und CI lassen sich mit dem Befehl INSTAT abfragen.

3) Das Eingangssignal CI kann den Computer zur Ausführung einer Interrupt-Routine veranlassen, und zwar unabhängig davon, ob er nun an- oder ausgeschaltet ist. Ist er ausgeschaltet, wird er durch das CI-Signal eingeschaltet. Beachten Sie hierzu die Erklärungen der Befehle ON PHONE GOSUB, PHONE ON/OFF/STOP und WAKE$ (s.Kapitel 14).

d) RS-232C-Befehle

Folgende Befehle sind im Zusammenhang mit dem RS-232C-Interface, das mit dem Parameter COM1: angesprochen wird, nutzbar:

COM$	Liefert einen String mit den eingestellten Werten, die über SETCOM vereinbart wurden.
COMn ON/OFF/STOP	Erlaubt oder verbietet Interrupts, die über das Interface angefordert werden.
INIT	Legt die Größe des Empfangspuffers fest.
INSTAT	Liefert die Werte der Steuersignale.
ON COMn GOSUB	Verzweigt in eine Interrupt-Routine, sofern eine Anforderung an der seriellen Schnittstelle anliegt.
ON PHONE GOSUB	Verzweigt in eine Interrupt-Routine, sofern auf der Telefon-Leitung eine entsprechende Anforderung anliegt.
OUTSTAT	Setzt die Pegel der Steuersignale gezielt auf "high" oder "low".
PCONSOLE	Definiert die Zeilenlänge und den EOL-Code.
PHONE ON/OFF/STOP	Erlaubt oder verbietet Interrupts, die von der Telefon-Leitung angefordert werden.
PZONE	Setzt die Druck-Zone für Ausgaben mit dem LPRINT-Befehl.
RCVSTAT	Setzt das Empfangsprotokoll sowie die Wartezeit (timeout).
SETCOM	Bestimmt das Kommunikationsprotokoll.
SETDEV	Spezifiziert das RS-232C-Interface für eine Ein- oder Ausgabe.
SNDBRK	Sendet Unterbrechungs-Codes, um den Datentransfer zu stoppen.
SNDSTAT	Setzt das Sendeprotokoll und die Wartezeit (timeout).

e) ERROR-Codes bei Verwendung des RS-232C-Ports

Die nachfolgende Auflistung beschreibt die ERROR-Codes, die beim Betrieb mit den beiden Schnittstellen COM1: (RS-232C) oder COM2: (SIO) auftreten können. Anhang F enthält sämtliche ERROR-Codes.

Code	Beschreibung
140	Ungültiger Parameter im SETCOM-Befehl.
141	Die mit INIT gewünschte Größe des Empfangspuffers überschreitet den maximal zulässigen Wert von 16383 Bytes oder den zur Verfügung stehenden Speicherplatz.
142	Fehler bei der Datenübertragung (falsche Parität, Überlauf, Einrahmungsfehler, Empfangspuffer voll).
143	Timeout-Error. Es erfolgte während der vorgegeben Zeit, die über RCVSTAT oder SNDSTAT vereinbart worden ist, keine Antwort vom angeschlossenen Gerät.
144	Der mit SETDEV spezifizierte serielle Port ist noch geöffnet.

B) Serieller Optischer Port (SIO-Interface)

Ihr PC-1600 unterstützt die serielle Datenübertragung auch auf optischem Wege. Hierfür ist eine mit SIO bezeichnete 5-polige Schnittstelle auf der rechten Seite des Computers angebracht.

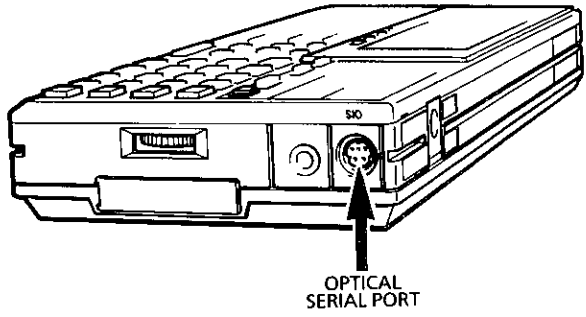


Abbildung 30 : SIO-Port

Dieser Abschnitt beschreibt die Anschlußbelegung und nennt die zuständigen BASIC-Befehle, die für die Bedienung dieses Ports gedacht sind. Darüberhinaus werden einige wichtige Punkte für den Umgang mit dieser Schnittstelle aufgezeigt.

Der Gebrauch der optoelektronischen Schnittstelle ist ähnlich dem des RS-232C-Ports, bietet aber einige Vorteile. Der wesentlichste ist dabei die hohe Übertragungsrate von 38400 baud im Halb-Duplex-Betrieb.

Eine optische Datenübertragung bedeutet aber die Notwendigkeit eines Glasfaser-Kabels, welches jedoch durch seine elektrische Isolierung eine störungsfreie Übertragung ermöglicht.

In den BASIC-Kommandos wird die optische Schnittstelle über den Parameter COM2: angesprochen, während für das RS-232C-Interface der Parameter COM1: gilt. Es kann immer nur einer dieser beiden Ports bedient werden, niemals jedoch beide gleichzeitig.

a) Hinweise zur Benutzung des SIO-Ports

1) Ausschaltung

Schalten Sie den Computer aus, wenn Sie ein externes Gerät an die optoelektronische Schnittstelle (SIO) anschließen oder hiervon entfernen.

2) Handhabung

Berühren Sie niemals die Schnittstellen-Kontakte am Computer, am Peripheriegerät oder am Kabel. Decken Sie alle Anschlüsse und Kontake mit den dafür vorgesehenen Kappen ab, sofern Sie diese nicht benötigen.
Das Glasfaserkabel kann zerstört werden, wenn sie es zu stark biegen. Die Folge sind Fehler in der Datenübertragung. Legen Sie das Glasfaserkabel daher so hin, daß möglichst weiche Rundungen entstehen und Knicke vermieden werden.

3) Kommunikations-Protokoll

Ein Datenaustausch über Glasfaserkabel zwischen dem PC-1600 und anderen Geräten ist nur dann realisierbar, wenn diese mit kompatiblen Parametern arbeiten. Diese Parameter lassen sich mit geeigneten BASIC-Befehlen einstellen, die auf einer der nächsten Seiten zusammengestellt sind.

4) Geräteanschluß

Vergewissern Sie sich, daß auch tatsächlich ein Gerät am SIO-Interface angeschlossen ist, bevor Sie einen Datenaustausch starten. Das Senden von Daten zu einem nicht angeschlossenen Peripheriegerät hat zur Folge, daß der Computer "abstürzt". Betätigen Sie in einem solchen Fall die [ON] (BREAK)-Taste, um einen Reset ohne Datenverlust durchzuführen.

5) Referenz-Hinweise

- Ziehen Sie die Bedienungsanleitungen des Glasfaserkabels und des angeschlossenen Gerätes (anderer Computer usw.) zu Rate, wenn Sie Informationen benötigen über:

  - die Kommunikations-Parameter,
  - die Kabelspezifikationen,
  - die Anschlußbelegungen und
  - die Sende- und Empfangsprozeduren.

- Schauen Sie in diesem Bedienungshandbuch nach, wenn Sie Informationen wünschen über:

  - die zuständigen BASIC-Befehle,
  - das Kommunikations-Protokoll des PC-1600,
  - die Kommunikations-Parameter des PC-1600,
  - die Anschlußbelegung des SIO-Ports und
  - die ERROR-Codes.

b) Standardeinstellungen des SIO-Ports

Die nachfolgende Liste zeigt Ihnen die Parameter-Einstellungen, die dem optoelektronischen Port (COM2:) vom System standardmäßig zugewiesen werden. Diese können mit den BASIC-Befehlen PCONSOLE und SETCOM geändert werden.

Befehl	Parameter	Standard-Wert
SETCOM	Übertragungsrate	38400 baud
	Wortlänge (5 bis 8 Bits)	7
	Parität (gerade, ungerade, keine)	gerade
	Stop-Bits (1 oder 2)	2
	X-ON/OFF-Protokoll	ja
	SHIFT in/out Protokoll (nur mit 7 Bit Wortlänge)	ja
PCONSOLE	Zeilen-Länge	unbegrenzt
	Zeilenende-Kennung (end of line code)	<CR> + <LF>

c) Anschlußbelegung des SIO-Ports

Folgende Abbildung zeigt das optionale Glasfaserkabel CE-1600L und die Numerierung des 5-Pol-Anschlusses der optoelektronischen Schnittstelle (SIO). Die Belegung dieser Anschlüsse wird in der anschließenden Tabelle aufgezeigt.

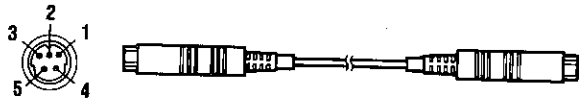


Abbildung 31 : Glasfaserkabel CE-1600L

Tabelle : Anschlußbelegung und Signal-Beschreibungen

Pin	Signal	Symbol	Kurzbeschreibung
1	Receive Data (Eingang)	RD	Empfängt serielle Daten, die von einem externen Gerät zum PC-1600 geliefert werden.
2	Ground	GND	Masse-Anschluß
3	Send Data	SD	Sendet serielle Daten zu einem externen Gerät.
4/5	Logic Voltage (Ausgang)	Vcc	Liefern die Versorgungsspannung für die externe Logik, wenn der Computer eingeschaltet ist. (Vcc = 4....5,5V)

d) Befehle für das SIO-Interface

Die nachfolgenden Befehle dienen der Benutzung der seriellen optoelektronischen Schnittstelle, die mit dem Parameter COM2: angesprochen wird. Eine vollständige Beschreibung dieser Befehle finden Sie in Kapitel 14.

COM$	Liefert einen String, der Auskunft über die mittels SETCOM eingestellten Kommunikations-Parameter gibt.
COMn ON/OFF/STOP	Erlaubt bzw. unterbindet das Wirksamwerden empfangener Interrupt-Anforderungen.
INIT	Definiert die Größe des Empfangspuffers.
ON COMn GOSUB	Erzwingt die Fortführung des Programmes mit der spezifizierten Unteroutine, sobald eine Interrupt-Anforderung empfangen wird.
PCONSOLE	Bestimmt Zeilenlänge und Zeilenend-Code.

PZONE	Bestimmt die Druckzone für Ausgaben mit dem LPRINT-Befehl.
RCVSTAT	Setzt die Wartezeit (timeout), innerhalb der etwas empfangen worden sein muß.
SETCOM	Bestimmt das Übertragungs-Protokoll.
SETDEV	Spezifiziert den optoelektronischen Port als Ein- bzw. Ausgang.
SNDBRK	Sendet Unterbrechungs-Codes, um die Datenübertragung zu stoppen.
SNDSTAT	Setzt die Wartezeit (timeout), innerhalb der die gesendeten Daten vom empfangenden Gerät als akzeptiert gemeldet werden müssen.

e) ERROR-Codes bei Verwendung des SIO-Ports

Die Liste der ERROR-Codes in den vorangegangenen Beschreibungen der RS-232C-Schnittstelle deckt auch die Fehler ab, die auch bei Verwendung des optoelektronischen Ports auftreten können. Eine Zusammenfassung aller ERROR-Codes ist in Anhang F gegeben.

6.4 Analoger Eingabe-Port

Ihr PC-1600 kann auch analoge Signale empfangen, wie sie unter anderem von Sensoren, Barcode-Lesern und dergleichen geliefert werden. Zu ihrem Empfang ist der mit ANALOG IN gekennzeichnete Anschluß vorgesehen, der sich auf der rechten Seite des PC-1600 befindet.

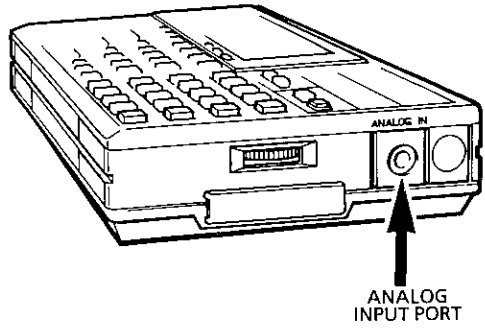


Abbildung 32 : Analoger Eingang

Dieser Abschnitt beschreibt die Spezifikationen dieses analogen Eingangs und listet die zu seiner Nutzung bereitstehenden BASIC-Befehle auf. Mit Hilfe dieser Befehle können analoge Signale in digitale Entsprechungen umgewandelt werden, um analoge Meßwerte vom Computer erfaß- und auswertbar zu machen. Durch vereinbarte Analogpegel lassen sich zudem Interrupt-Routinen starten.

a) Elektrische Kennwerte des Analog-Ports

Der effiziente Gebrauch des analogen Eingabe-Ports hängt von der Kombination der analogen Signalquelle und der Software-Steuerung in der Programmiersprache BASIC oder der Maschinensprache ab, um eine getreue Umwandlung in das digitale Äquivalent zu erhalten. Die nachfolgende Tabelle nennt die elektrischen Kennwerte dieses analogen Port, d.h. seine Spezifikationen.

Spezifikationen des Analog-Einganges

Maximal zulässige Eingangs-Spannung	:	4 V
Bereich normaler Eingangs-Spannungen	:	0 bis 2,495 V
Wertebereich der digitalen Wandlung	:	0 bis 255 (8-Bit-Wert) (Spannungen > 2,495 V werden in den Wert 255) gewandelt)
Wandlungsfehler	:	3% ± 2 Ziffern (digits)
Eingangsimpedanz	:	100 kΩ ± 2%

WICHTIG: Um eine Zerstörung von Computer und Analog-Eingang zu vermeiden, sollte die Eingangsspannung innerhalb des Bereiches von 0 bis 4V gehalten werden.

b) Befehle für den analogen Eingang

Mit den nachstehend aufgeführten Befehlen kann der Analog-Port zum Empfang von Daten genutzt werden. Eine genaue Beschreibung dieser Befehle ist in Kapitel 14 zu finden.

ADIN ON/OFF/STOP	Erlaubt bzw. verhindert die Durchführung von Interrupts, die über den Analog-Eingang angefordert werden.
AIN	Liefert einen Dezimalwert zwischen 0 und 255, der dem analogen Eingangspegel entspricht.

ON ADIN GOSUB — Erzwingt die Fortführung des Programmes mit der durch GOSUB definierten Routine, sofern die analoge Eingangsspannung innerhalb eines vereinbarten Bereiches liegt.

6.5 Drucker mit Cassetten-Recorder-Interface

Der Drucker CE-1600P befähigt Ihren Taschencomputer zur Ausgabe von vierfarbigen Texten und Grafiken. Er stellt desweiteren die Grundlage für den Anschluß weiterer Optionen dar.

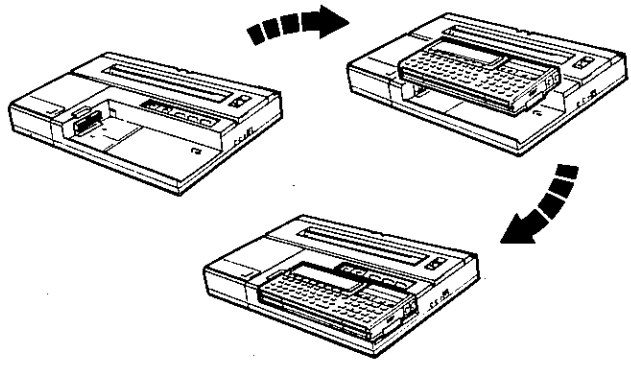


Abbildung 33 : Anschluß an den Drucker CE-1600P

Dieser Abschnitt beschreibt die wichtigsten Punkte, die Sie beim Umgang mit dem Drucker und seinem Recorder-Interface beherzigen sollten. Er listet auch die BASIC-Befehle und ERROR-Codes auf, die im Zusammenhang mit diesem Gerät stehen.

Der Drucker CE-1600P ist so gestaltet, daß der Taschencomputer PC-1600 in einer Führungsschiene so in diesen integriert werden kann, daß eine Einheit von ebenmäßigem und ansprechendem Profil entsteht. Das optionale Diskettenlaufwerk läßt sich in ähnlicher Weise in den Drucker integrieren und komplettiert damit Ihren PC-1600 zu einem kompakten System. Durch die direkte Verbindung von Drucker und Computer über den Systembus sowie eine ähnlich einfache Integrierbarkeit des Diskettenlaufwerkes erübrigt sich der Gebrauch separater Kabel.

Der Drucker wird über eingebaute Akkus (NiCad-Zellen) betrieben, die sich mit dem mitgelieferten Netzadapter (EA-160) wieder aufladen lassen. Sie versorgen auch ein angeschlossenes Disketten-Laufwerk sowie den Computer selbst, sofern dieses erforderlich ist.

Wenn Sie den Drucker zu Hause oder im Büro benutzen, sollten Sie ihn über den standardmäßig mitgelieferten Netzadapter (EA-160) am Wechselstromnetz betreiben, um die Akkus zu schonen.

a) Hinweise zum Gebrauch des Druckers:

1) Einschaltung

Der Drucker besitzt keinen Ein-/Ausschalter. Ist er mit dem Computer verbunden, so wird er zusammen mit dem Computer über dessen Taste ON eingeschaltet. Schalten Sie den Computer und damit den Drucker aus, bevor Sie beide Geräte zusammenbringen oder voneinander trennen.

2) Handhabung

Berühren Sie niemals die Anschlußkontakte des Druckers oder des Systembusses des Computers. Erden Sie sich gegebenenfalls an metallischen Objekten bzw. entladen Sie sich an diesen, sofern elektrostatische Ladungen zu einem Problem werden. Schützen Sie die Kontakte mit den dafür vorgesehenen Kappen und Abdeckungen, wenn Sie nicht in Gebrauch sind.

3) Entladene Akkus

Wenn Sie den Drucker zum ersten Male benutzen, sollten Sie ihn über den Netzadapter betreiben, um seine eingebauten Akkus aufzuladen. Entladene Akkus meldet der PC-1600 auf dreierlei Art:

- Anzeige des Symboles BATT in der Statuszeile
- Anzeige einer CHECK-Meldung nach dem Einschalten
- Anzeige eines ERROR-Codes während des Betriebes

4) RESET-Schalter

Ist der Computer in die Drucker-Einheit integriert, besteht nicht mehr die Möglichkeit, an den auf seiner Unterseite befindlichen RESET-Schalter heranzukommen. Um dennoch einen Reset erzeugen zu können, ist auf der Unterseite des Druckers ebenfalls ein RESET-Schalter angebracht, der dieselbe Wirkung hat wie der im Computer eingebaute Schalter.

5) Recorder-Interface

Das eingebaute Recorder-Interface erlaubt den Anschluß eines optionalen Cassetten-Recorders (z.B. SHARP CE-152). Befindet sich der REMOTE-Schalter des Druckers in Stellung ON, kann der Recorder-Motor durch die BASIC-Befehle, die im Zusammenhang mit Cassetten-Dateien stehen, fernbedient werden. Lesen Sie hierzu bitte auch den nächsten Abschnitt: "Gebrauch eines Cassetten-Recorders".

6) Referenz-Hinweise

- o Aus der Bedienungsanleitung des Druckers CE-1600P erhalten Sie Informationen über:
  - Details des Druckergebrauchs,
  - Vorsichtsmaßnahmen im Umgamg mit dem Drucker,
  - Stiftwechsel,
  - Batterie-Aufladung,
  - Papierzufuhr und Justage,
  - Allgemeine Spezifikationen und
  - Druckerzubehör.
- o Aus diesem Handbuch bekommen Sie Informationen über:
  - druckerrelevante BASIC-Befehle,
  - ERROR-Codes und
  - Grafik- und Text-Modus-Einstellungen.

b) Befehle für den Drucker

Mit nachstehend gezeigten Befehlen können Daten an den Drucker übergeben sowie Druckformate und Druck-Modi eingestellt werden. Eine vollständige Beschreibung dieser Befehle ist aus Kapitel 14 ersichtlich.

COLOR	wählt den Farbstift aus.
CSIZE	bestimmt die Größe der zu druckenden Zeichen.
GLCURSOR	setzt den Stift an eine bestimmbare Grafik-Position.
GRAPH	schaltet den Drucker in den Grafik-Modus.
LCURSOR	bewegt den Stift an die spezifizierte Text-Spalte.
LF	bewirkt Papiervorschub um gewünschte Zeilenanzahl.
LLINE	zeichnet eine Gerade zwischen absoluten Koordinaten.
LLIST	listet ein Programm auf dem Papier auf.
LLIST*	listet REMark-Zeilen auf dem Papier auf.
LPRINT	leitet Daten an den Drucker weiter.
PAPER	bestimmt den Papiertyp und vertikalen Druckbereich.
PCONSOLE	setzt das Druckformat und den Zeilenend-Code (EOL).
PITCH	bestimmt Zeichen- und Zeilenabstand im Text-Modus.

PZONE	spezifiziert die Druckzone für den LPRINT-Befehl.
RLINE	zeichnet eine Gerade zwischen relativen Koordinaten.
ROTATE	bestimmt die Druckrichtung von Lage der Zeichen.
SORGN	bestimmt die momentane Stiftposition als Ursprung des Koordinatensystemes.
TAB	bewegt den Stift an eine bestimmbare Text-Spalte.
TEST	führt einen Test mit den vier Farbstiften durch.
TEXT	versetzt den Drucker in den Text-Modus.

c) ERROR-Codes bei Verwendung des Druckers

Diese ERROR-Code-Liste bezieht sich auf Fehler, die bei einer Verwendung des Druckers auftauchen können. Eine Zusammenfassung aller ERROR-Codes ist in Anhang F zu finden.

Code	Beschreibung
70	Stift außerhalb des Bereiches X=-2048, Y=2047.
71	Papierrückführung im Text-Modus größer als 10,24 cm. (Gilt nur für CE-150.)
72	Unzulässige Parameter-Angabe beim TAB-Befehl.
73	Unerlaubter Befehl im Grafik- oder Text-Modus.
74	Zu viele Parameter in der LINE- oder RLINE-Anweisung.
76	Mit LLIST aufzulistende Programmzeile ist zu lang, um bei der mit PCONSOLE gewählten Einstellung dargestellt werden zu können.
78	Schwache Batterie: Stift nicht am richtigen Platz oder Druckwerk blockiert. LPRINT- bzw. LINE-Befehl kann nicht ausgeführt werden.
79	Farbwahl nicht an den Drucker weitergeleitet. (Gilt nur bei CE-150.)
80	Schwache Batterie: Drucker hat sich "aufgehängt".

6.6 Gebrauch eines Cassetten-Recorders

Über die dafür eigens vorgesehene Schnittstelle können Sie am Drucker CE-1600P einen Cassetten-Recorder (z.B. SHARP CE-152) anschließen. Die nachfolgende Abbildung zeigt die Verbindung dieser beiden Geräte über ein Kabel.

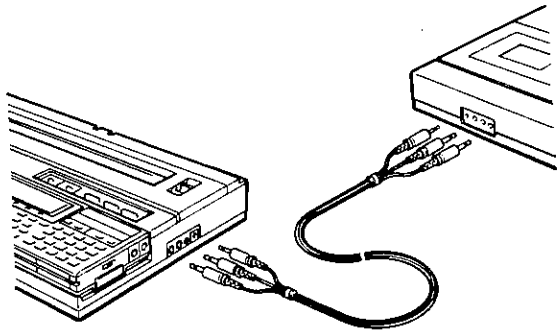


Abbildung 34 : Anschluß eines Recorders

Dieser Abschnitt beschreibt die BASIC-Befehle, die dem Schreiben und Lesen von Daten auf bzw. vom Recorder dienen und zeigt eine Liste der möglichen ERROR-Codes sowie einige wichtige Hinweise für den Anschluß und richtigen Umgang mit einem Recorder.

a) Hinweise zum Gebrauch eines Recorders:

1) Einschaltung

Stellen Sie sicher, daß der Computer ausgeschaltet ist, bevor Sie eine Verbindung zum Cassetten-Recorder herstellen.

2) Kabelverbindung

Der Drucker CE-1600P hat drei Klinken-Buchsen auf der rechten Gehäuseseite. Hieran kann der Recorder angeschlossen werden. Diese Buchsen tragen die Bezeichnungen: REM, MIC und EAR und haben folgende Bedeutung:

REM (remote)	Fernbedienung des Recorder-Motors
MIC (microphone)	Signaleingang
EAR (earphone)	Signalausgang

Das mit dem Recorder CE-152 mitgelieferte Kabel ist farblich codiert, so daß ein problemloser Anschluß gewährleistet sein sollte. Beachten Sie, daß die beiden Stecker für die MIC- und EAR-Buchse dieselbe Größe haben und versehentlich vertauscht gesteckt werden können. Überprüfen Sie anhand der farblichen Kennzeichnung, ob ein richtiger Anschluß vorliegt.

3) Fernbedienung

Stellen Sie den REMOTE-Schalter des Druckers in Stellung ON, wenn Sie möchten, daß sich über die Befehle RMT ON und RMT OFF die Fernbedienbarkeit des Recorders ein- oder ausschalten läßt.

4) Verwendung SHARP-fremder Recorder

Benutzen Sie einen anderen Cassetten-Recorder als den CE-152, so muß dieser eine zum PC-1600 kompatible Ausgangsspannung liefern. Anderenfalls wird der ERROR-Code 44 angezeigt, wenn Daten von diesem Recorder gelesen werden sollen. Tritt ein solcher Fall ein, versuchen Sie diesem Umstand mit folgender Anweisung zu begegnen:

POKE &F1A9,0 [ENTER]

WICHTIG: Seien Sie besonders vorsichtig bei dieser Eingabe, da Sie hiermit auf die Systemebene des Computers zugreifen. Bevor Sie die ENTER-Taste betätigen, vergewissern Sie sich lieber nochmals, ob auch die richtige POKE-Adresse und das richtige Argument von Ihnen eingetippt worden ist. Ein Fehler in dieser Anweisung kann nach Aktivierung derselben nur durch einen Total-Reset behoben werden. Alle Programme und Daten gehen damit natürlich verloren. Wenn Sie nach einer Falscheingabe keinen Reset durchführen, erscheint der ERROR 44 beim nächsten Versuch den Recorder zu benutzen wieder.

5) Referenz-Hinweise

- Nehmen Sie die Bedienungsanleitungen des Druckers CE-1600P und des Recorders CE-152 (oder eines anderen Recorders) zur Hand, um sich zu informieren über:
  - das Recorder-Anschlußkabel und dessen Anschlußweise,
  - die Spezifikationen des Recorders,
  - grundlegende Vorsichtsmaßnahmen und Wartungshinweise,
  - und Bedienungs-Prozeduren.
- Ziehen Sie dagegen dieses Handbuch zu Rate, wenn Sie etwas wissen möchten über:
  - die BASIC-Befehle, mit denen Programme bzw. Daten auf Cassette aufgezeichnet und von dieser wieder gelesen werden können und
  - die Bedeutung der ERROR-Codes, die im Zusammenhang mit der Benutzung des Recorders auftreten können.

b) Recorder-Befehle

Mit den nachstehenden BASIC-Befehlen lassen sich Daten mit Hilfe eines Recorders aufzeichnen, die Aufzeichnung auf Fehlerfreiheit überprüfen, Daten vom Recorder in den Computer laden sowie der Motor des Recorders ein- bzw. ausschalten. Diese Befehle sind in Kapitel 14 ausführlich erläutert.

CHAIN	Gestattet einem BASIC-Programm den Aufruf und die Durchführung eines anderen BASIC-Programmes.
CLOAD	Lädt ein BASIC-Programm oder eine Funktionstastenbelegung.
CLOAD?	Überprüft, ob Cassetten-Datei und Speicherinhalt des Computers identisch sind, ob also die Datei fehlerfrei aufgezeichnet oder geladen worden ist.
CLOADM	Lädt ein Maschinensprache-Programm von Cassette in den Speicher des Computers.
COPY	Kopiert eine Datei von Cassette zu einem externen Gerät oder umgekehrt von diesem auf Cassette.
CSAVE	Zeichnet ein BASIC-Programm auf Cassette auf.
CSAVEM	Zeichnet ein Maschinensprache-Programm auf einer Cassette auf.
EOF	Liefert einen Wert, der das Ende einer sequentiell vom Band geladenen Datei signalisiert.
INPUT#	Liest den nächsten Datensatz einer auf Cassette befindlichen sequentiellen Datei.
MERGE	Kombiniert zwei auf Cassette befindliche Programme im Arbeitsspeicher zu einem Gesamtprogramm.
PRINT#	Schreibt einen Datensatz sequentiell in die Datei, die auf Cassette aufgezeichnet wird.
RMT ON/OFF	Schaltet die automatische Steuerung des Recorders ein bzw. aus.

c) ERROR-Codes bei Verwendung eines Recorders

Die nun folgende Liste bezieht sich auf jene ERROR-Codes, die im Zusammenhang bei der Verwendung der eben genannten BASIC-Befehle auftreten können. Eine vollständige Beschreibung aller möglichen ERROR-Codes ist in Anhang F zusammengestellt.

Code	Beschreibung
40	Syntax-Fehler in einer recorderbezogenen Anweisung.
42	Nicht genügend Speicherplatz zum Laden des Programmes.
43	Überprüfung negativ. Die auf Cassette befindliche Datei stimmt nicht mit dem Arbeitsspeicher-Inhalt überein. Unter Umständen wurde das Programm fehlerhaft vom Band geladen. Laden Sie es nochmals und führen Sie erneut einen Vergleich mit CLOAD? durch. Möglicherweise liegt auch eine falsche Typenangabe in der INPUT#-Anweisung vor. Der Typ der Variable und der aus der Datei zu holende Wert stimmen nicht überein. Es wird versucht, ein String in eine numerische Variable zu laden oder umgekehrt.
44	Inkompatible Ausgangsspannung des Recorders.

6.7 Disketten-Laufwerk

Die Hinzufügung des Disketten-Laufwerks CE-1600F komplettiert die Kombination aus Drucker und Computer zu einem kompakten und noch leistungsstärkeren System.

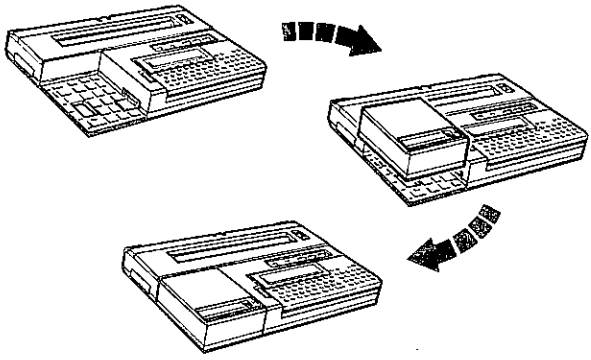


Abbildung 35 : Anschluß des Disketten-Laufwerkes

Dieser Abschnitt beschreibt die Punkte, die beim Anschluß eines solchen Laufwerkes beachtet werden sollten, und zeigt, welche BASIC-Befehle zu dessen Bedienung zur Verfügung stehen und was für ERROR-Codes im Zusammenhang mit ihrer Benutzung im Fehlerfalle auftreten können.

Auf den mit dem Laufwerk verwendbaren Disketten (pocket disks) lassen sich pro Seite bis zu 61 KByte speichern. Sie sind damit ein effizientes Werkzeug zur Massenspeicherung von Daten, die nicht in RAM-Modulen oder sinnvoll auf Cassette untergebracht werden könnten. Trotz ihres kleinen 2.5"-Formates besitzen sie eine recht hohe Speicherkapazität. Mit geeigneten BASIC-Befehlen lassen sie sich ebenso einfach handhaben wie Disketten größeren Formates.

Das Diskettenlaufwerk selbst wird direkt in die Drucker-Einheit CE-1600P eingesetzt und bedarf somit keiner Verbindungskabel. Sind Computer und Disketten-Laufwerk in den Drucker integriert, so ergibt sich ein einheitlich gestaltetes Gesamtbild.

a) Hinweise zum Gebrauch des Disketten-Laufwerks:

1) Ausschaltung

Schalten Sie den Computer aus, bevor Sie das Laufwerk in die Drucker-Einheit integrieren. Berühren Sie auf keinen Fall die Anschlußkontakte des Laufwerkes und schützen Sie diese bei Nichtverwendung stets mit den dafür vorgesehenen Abdeckungen.

2) Disketten-Formatierung

Bevor sich neue Disketten benutzen lassen, müssen diese erst formatiert werden. Da die Disketten für doppelseitige Datenaufnahme ausgelegt sind, muß diese Formatierung für jede der beiden Seiten separat geschehen. Es wird dabei immer auf die Seite zugegriffen, die nach oben zeigt. Um auch die andere Seite formatieren zu können, ist die Diskette zu wenden und erneut eine Formatierungsanweisung einzugeben. Eine solche Anweisung sieht folgendermaßen aus: INIT "X:"

Vergessen Sie nicht, diese Eingabe durch Betätigung der Taste ENTER zu aktivieren. Das Formatieren einer Disketten-Seite dauert weniger als eine Minute. Erlischt das grüne Lämpchen, kann die Diskette entnommen, umgedreht wieder eingelegt und die andere Seite formatiert werden.

3) Handhabung der Disketten

Während eines Diskettenzugriffs leuchtet ein grünes Lämpchen, das sich auf der Oberseite des Laufwerkes befindet, auf. In dieser Phase darf die Diskette keinesfalls aus dem Laufwerk herausgenommen werden, da hierdurch ihr Dateninhalt oder aber auch die Diskette selbst zerstört werden können. Diese Gefahr besteht ebenso für den Laufwerksmechanismus. Entnehmen Sie ebenfalls keine Diskette, solange sich auf dieser noch eine Datei im geöffneten Zustand befindet.

4) Versorgungsspannung

Das Disketten-Laufwerk hat keine eigene Spannungsversorgung, wird aber über die wiederaufladbaren Akkus des Druckers mitversorgt. Sobald in der Statuszeile das BATT Symbol auf entladene Akkus hinweist, sollten Sie den Computer über einen Netzadapter betreiben, um diese wieder aufzuladen. Benutzen Sie das Disketten-Laufwerk trotz dieser warnenden Anzeige, so muß mit dem Verlust von Daten gerechnet werden.

5) Dateien

Die Disketten CE-1650F können bis zu 48 Dateien oder 61 KByte pro Seite aufnehmen. Sobald der Computer durch die Anweisung POWER OFF oder die Betätigung der [OFF] Taste ausgeschaltet wird, schließen sich alle geöffneten Dateien automatisch.

6) Sicherheitskopien (backup files)

Die im SHARP PC-1600 Taschencomputer verwendete BASIC-Version erkennt zwei logische Gerätebezeichnungen für das Disketten-Laufwerk, nämlich: X: und Y:. Die Standardbezeichnung ist X:. Der Gebrauch dieser beiden logischen Bezeichner für ein und dasselbe Laufwerk gestattet es, von den wichtigsten Dateien Sicherheitskopien auf anderen Disketten anfertigen zu können. Dabei kann immer nur eine Datei zur Zeit kopiert werden. Die Verwendung mehrdeutiger Dateibezeichnungen durch den Gebrauch von Jokerzeichen (wildcards) ist dabei ausgeschlossen. Ebenso muß in jeder Dateibezeichnung die Extension angegeben sein. Angenommen Sie haben eine Diskette im Laufwerk und letzteres als Gerät X: spezifiziert. Um nun eine hierauf befindliche Datei auf eine andere Diskette zu kopieren, geben Sie einfach die Anweisung:

COPY "X:<Dateiname>.<Ext>" TO "Y:<Dateiname>.<Ext>"

ein und betätigen die ENTER-Taste. Dieses teilt dem Computer mit, daß die gewünschte Datei der Diskette X: auf Diskette Y: zu kopieren ist. Da sich die Bezeichnungen X: und Y: auf dasselbe Laufwerk beziehen, fordert Sie der Computer, nachdem er die Datei von Diskette X: in den Speicher geladen hat, auf, eine andere Diskette in das Laufwerk einzulegen. Hierzu wird auf dem Display folgende Meldung angezeigt:

Set diskette for Y: (könnte auch X: lauten)

Sobald diese Meldung erscheint, entfernen Sie die im Laufwerk befindliche Diskette, setzen eine andere Diskette ein und betätigen anschließend die [ENTER] -Taste. Die in dem Arbeitsspeicher stehende Original-Datei wird damit auf die zweite Diskette kopiert.

Da der Computer nur blockweise kopieren kann, mag es bei sehr umfangreichen Dateien vorkommen, daß Sie zu einem mehrfachen Wechsel zwischen Quell- und Zieldiskette aufgefordert werden. Ist der Sicherungsvorgang abgeschlossen, stoppt das Laufwerk und die grüne Lampe erlischt. Ebenso erscheint dann auch das Bereitschaftszeichen > wieder auf dem Display. Sie können nun die Sicherungsdiskette entnehmen, aufbewahren und den PC-1600 für andere Dinge nutzen.

Weitergehende Informationen erhalten Sie aus den Erklärungen der Befehle COPY und FILES, die im Kapitel 14 gegeben sind.

7) Referenz-Hinweise

- o Die Bedienungsanleitungen des Druckers CE-1600P und des Disketten-Laufwerks CE-1600F liefern Ihnen Informationen über:

  - Anschluß des Laufwerkes,
  - Einlegen und Entnahme von Disketten,
  - Schutz der Disketten vor versehentlichem Beschreiben,
  - Spezifikationen und
  - Grundsätzliche Vorsichtsmaßnahmen und Bedienungshinweise

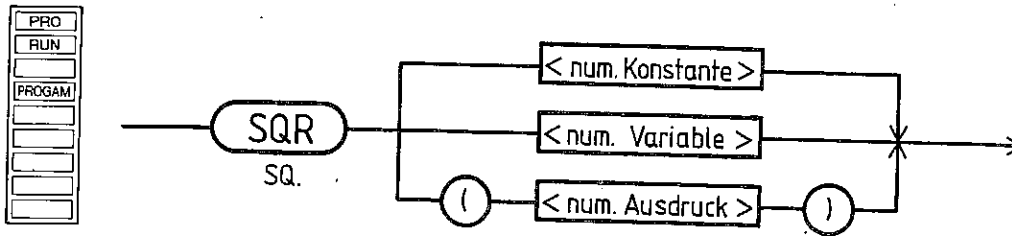
- o Dieses Handbuch gibt Auskunft über:

  - die BASIC-Befehle zum Schreiben und Lesen von Daten,
  - die dabei möglicherweise auftretenden ERROR-Codes und
  - die Erzeugung und Änderung von Dateien in BASIC.

b) Disketten-Befehle

Die im folgenden aufgelisteten BASIC-Befehle erlauben Ihnen das Lesen von Disketten-Dateien oder das Schreiben derselben. Genaue Erklärungen hierzu finden Sie in Kapitel 14.

BLOAD	Lädt ein Maschinensprache-Programm von der Diskette in den Arbeitsspeicher des Computers.
BSAVE	Schreibt ein Maschinensprache-Programm auf Diskette.
CLOSE	Schließt eine Datei und beendet so die Möglichkeit, auf diese zugreifen zu können.
COPY	Kopiert eine Datei unter anderem Namen auf demselben Speichermedium oder unter beliebigem Namen von einem Medium auf ein anderes.



WIRKUNG : Die Funktion SQR(X) liefert den positiven Wert der Quadratwurzel von X.

HINWEISE : Das Argument X darf ein beliebiger numerischer Ausdruck sein, dessen Wert jedoch im positiven Zahlen-Bereich liegen oder Null lauten muß. Negative Werte sind nicht erlaubt und führen daher zur Ausgabe eines ERROR-Codes.

Anstelle des funktionalen Operators SQR ist auch die Verwendung des Wurzelzeichens √ möglich. Dieses Zeichen ist in MODE 1 durch die offene eckige Klammer [ersetzt.

BEISPIELE :

```
>
SQR(5)
                    2.236067977
>


>
MODE 1
√2
                    1.414213562
>


>
MODE 0
[3
                    1.732050808
>
```

c) ERROR-Codes bei Verwendung des Disketten-Laufwerkes

Die nachfolgende Liste zeigt jene ERROR-Codes, die in Verbindung mit der Benutzung des Diskettenlaufwerks auftreten können. Eine Zusammenfassung sämtlicher ERROR-Codes ist in Anhang F gegeben.

Code	Beschreibung
150	Zu große Dateienanzahl in der MAXFILES-Anweisung.
151	Datei existiert bereits. Wählen Sie einen anderen Namen.
152	Datei nicht gefunden. (Angegebenen Namen überprüfen.)
153	Falsche Dateinummer in der Lese- oder Schreibanweisung. Gewünschte Datei ist noch nicht geöffnet.
154	Datei ist bereits geöffnet. Sie muß geschlossen werden, bevor sie zu einem neuen Zweck geöffnet werden kann.
155	Unzulässige Laufwerksbezeichnung oder Laufwerk ist nicht angeschlossen.
156	Fehlerhafter Parameter in der SET-Anweisung.
157	Unzulässiger Dateiname bzw. fehlerhafte Spezifikation.
158	Befehl ist nicht auf das Laufwerk anwendbar.
159	Schreibversuch auf eine schreibgeschützte Diskette.
160	Keine Diskette im Laufwerk.
161	Diskette ist nicht mit INIT formatiert worden.
162	Schreib- oder Lesefehler.
163	Falsche Diskette im Laufwerk. Diskette wurde gewechselt während eine geöffnete Datei vorlag.
164	Diskette ist voll.
165	Dateiende mit der INPUT#-Anweisung erreicht. Alle Daten sind bereits gelesen.
166	Nicht genügend Speicherplatz im Computer frei, um das Laufwerk bedienen zu können.
167	Fataler Fehler: Diskette beschädigt oder Inhalt zerstört.
168	Laufwerk defekt oder Akkus des Druckers leer.

7. Benutzung der PC-1500-Optionen

Eine wesentliche Grundlage für die Flexibilität des PC-1600 ist unter anderem seine Kompatibilität zum PC-1500. Das bedeutet, daß er mit den meisten Peripheriegeräten und anderem Zubehör des PC-1500 betrieben werden kann. Diese Optionen seien nachfolgend kurz beschrieben. Bezüglich deren Lieferbarkeit sollten Sie sich an Ihren SHARP-Händler wenden, der Ihnen auch Tips zur Anwendung dieses Zubehörs geben kann. Beachten Sie vor dem Gebrauch einer Option bitte unbedingt die mitgelieferte Bedienungsanleitung, um Details, die hier nicht dargestellt werden können, zu erfahren.

In diesem Handbuch haben wir bereits auf einige dieser Optionen wie RAM-Module und Cassetten-Recorder hingewiesen. Beachten Sie, daß sich der PC-1600 genau wie ein PC-1500 verhalten, ihn also emulieren kann, wenn er sich im Betriebsmodus MODE 1 befindet. Weitere Informationen hierüber gibt Ihnen Kapitel 9.

Nehmen Sie bitte auch den Anhang H zur Kenntnis, der auf einige kompatible BASIC-Befehle eingeht und das Zubehör nennt, daß mit beiden Computern (PC-1500 und PC-1600) verwendbar ist.

Folgende PC-1500-Optionen sind auch mit dem PC-1600 benutzbar:

CE-150 **Drucker mit Cassetten-Recorder-Schnittstelle.**

Dieses Gerät stellt eine kleinere Version des Druckers CE-1600P dar. Es lassen sich hiermit ebenfalls Texte und Grafiken mit vier verschiedenen Farben erstellen. Das eingebaute Recorder-Interface erlaubt den Anschluß zweier Cassetten-Recorder.

CE-151 **RAM-Modul**

Erweitert den Arbeitsspeicher um 4 KByte.

CE-152 **Cassetten-Recorder**

Anschließbar an die Schnittstelle des CE-1600P oder des CE-150 Druckers. Der Recorder kann auch an die Einheit CE-162E (Parallel u. Recorder-Interface) angeschlossen werden.

CE-153 Programmierbare Tastatur

Mit dieser Tastatur werden weitere Funktionstasten, die in einer 10*14-Matrix angeordnet sind, bereitgestellt, um die Eingabe sich ständig wiederholender Funktionen zu vereinfachen.

CE-155 RAM-Modul

Erweitert den Arbeitsspeicher um 8 KByte.

CE-158 Seriell/Parallel-Schnittstellen-Einheit

Stellt ein serielles RS-232C-Interface für den Datenaustausch zwischen peripheren Geräten (z.B. Computer, Modem oder Drucker) zur Verfügung. Darüberhinaus ist über das Centronics-Interface ein Drucker anschließbar.

CE-159 RAM-Modul

8-KB-Modul mit Schreibschutz und Pufferungsbatterie.

CE-161 RAM-Modul

16-KB-Modul mit Schreibschutz und Pufferungsbatterie.

CE-162E Parallel- u. Cassetten-Recorder-Interface

Diese Einheit erlaubt den Anschluß eines Druckers mit Centronics-Interface und den eines Cassetten-Recorders.

EA-150 Netzadapter

Standardzubehör zum Drucker CE-150

EA-160 Netzadapter

Standardzubehör zum Drucker CE-1600P

TEIL IV

BASIC-REFERENZTEIL

Dieser Teil beschreibt die Aspekte, die im Zusammenhang mit der speziell adaptierten Version des SHARP-BASICs stehen. Sie sind die Basis für die außergewöhnliche Vielseitigkeit des Taschencomputers PC-1600.

Kapitel 8 gibt grundlegende Hinweise zur Erstellung von Programmen und deren Start sowie einen Überblick über die Belegung des internen Speichers.

Kapitel 9 zeigt, in welchen verschiedenen Modi der PC-1600 unter der Verwaltung des BASIC-Interpreters betrieben werden kann.

KAPITEL 10 nennt Ihnen eine Unterscheidung der Datentypen und zeigt deren Darstellung in Form von Konstanten und Variablen sowie deren Verknüpfung zu Ausdrücken unter der Anwendung von Operatoren auf.

Kapitel 11 beschreibt die Handhabung von Dateien, die auf einer Cassette, Diskette oder RAM-Disk gespeichert werden können.

Kapitel 12 handelt von der Benutzung der seriellen Schnittstellen sowie der damit verbundenen Kommunikation mit peripheren Geräten.

Kapitel 13 erläutert einfache Techniken zur Beseitigung von Fehlern.

Kapitel 14 erklärt alle vom PC-1600-BASIC bereitgestellten Befehle in alphabetischer Reihenfolge.

8. Grundlegendes zur Programmierung

Ein Programm ist eine logisch aufgebaute Folge von Anweisungen, die alle erforderlichen Operationen eines Computers steuern, um diesen zur Lösung einer Aufgabe zu befähigen. Diese Anweisungen müssen dazu in einer Sprache geschrieben sein, die der Computer "verstehen" kann. Da sämtliche Funktionen des PC-1600 von einem Prozessor gesteuert werden, kennt der Computer im Grunde nur die Befehlssprache dieses elektronischen Bausteines, die sogenannte Maschinensprache. Durch ein spezielles und residentes Programm wird der Computer jedoch befähigt, eine anwenderfreundlichere Sprache zu akzeptieren. Dieses Programm ist in der Lage, die Befehle der Programmiersprache BASIC zu interpretieren und in die erforderlichen Maschinen-Codes umzusetzen. Fortgeschrittenen Anwendern bleibt es jedoch unbenommen, auf den Prozessor direkt zuzugreifen und Maschinensprache-Programme ablaufen zu lassen.

Für die meisten Anwender wird jedoch der BASIC-Interpreter das wichtigste Instrument zur Handhabung eigener Programme bleiben. Das SHARP-BASIC des PC-1600 umfaßt einen Vorrat von etwa 200 Befehlen, die in Kommandos, Anweisungen und Funktionen unterschieden werden. Die Regeln, die die formale Anwendung dieser Befehle beschreiben, nennt man die Befehlssyntax. Diese Syntax läßt sich nicht immer mit einfachen Worten beschreiben, so daß wir diese Regeln bei der Erklärung der BASIC-Befehle in Form von Diagrammen wiedergegeben haben (s. auch Anhang K).

8.1 Eingabe von BASIC-Befehlen

Bei der Eingabe von BASIC-Befehlen lassen sich zwei verschiedene Betriebsarten voneinander unterscheiden: der direkte und der indirekte Modus. Im direkten Modus werden die in den Computer eingegebenen Befehlswörter unmittelbar nach der Betätigung der ENTER Taste analysiert und der betreffende Befehl ausgeführt. Nach Ausführung eines Befehles ist der Computer wieder bereit, weitere Befehle entgegenzunehmen.

Der DIREKTE MODUS kann mit der Betriebsart eines Taschenrechners verglichen werden. Er ist nützlich bei der Berechnung einfacher und nicht ständig wiederkehrender Berechnungen (s. Kapitel 5). Ebenso dient dieser Modus der Ausführung von Kommandos, die die Steuerung peripherer Geräte (z.B. des Diskettenlaufwerkes) und die Einstellung bestimmter Betriebsparameter vornehmen.

Das bedeutet aber nicht, daß BASIC-Befehle, die zur Gruppe der Kommandos gehören, nur im direkten Modus ausführbar wären. Einige Kommandos lassen sich auch programmieren, also indirekt abarbeiten. Ebenso gibt es Kommandos, die nur auf indirektem Wege ausgeführt werden können.

Wie auch immer, im direkten Modus kann stets nur ein Befehl zur Zeit abgearbeitet werden. Gibt man mehrere Befehle ein, bevor die Taste ENTER betätigt wird, erscheint eine Fehlermeldung auf dem Display (ERROR 1). In seltenen Fällen kann es aber auch passieren, daß der Computer nur den ersten Befehl akzeptiert und alle weiteren Befehle ignoriert.

Im PROgrammier-Modus werden die nacheinander auszuführenden Befehle in sogenannten Programm-Zeilen zusammengefaßt, wobei jede Zeile zu Beginn mit einer sogenannten Zeilen-Nummer zu versehen ist. Diese Numerierung legt fest, in welcher Reihenfolge die einzelnen Zeilen abzuarbeiten sind. Alle Programm-Zeilen werden im Computer gespeichert, aber nicht ausgeführt. Die Ausführung erfolgt erst auf ein spezielles Kommando, wie z.B. RUN.

Nach einem solchen Programm-Start beginnt der Computer mit der Abarbeitung der gespeicherten Befehle. Dazu beginnt er mit der Zeile, die die niederwertigste Nummer trägt und setzt seine Arbeit Schritt für Schritt mit den folgenden Zeilen fort, bis keine weitere Programm-Zeile im Speicher vorgefunden werden kann oder ein spezieller Befehl den Programmablauf stoppt.

Die Programm-Größe bzw. die Anzahl der im Computer aufnehmbaren Programmzeilen hängt von der Kapazität des Arbeitsspeichers ab.

Nachfolgend sei ein kleines Programm-Beispiel gezeigt, dessen Zeilen im Zehnerabstand numeriert worden sind. Diese Art der Numerierung ist eine übliche und gute Praxis, weil sich somit vergessene Zeilen später leicht in das Programm einfügen lassen.

```
10 PRINT "DIESES EINFACHE PROGRAMM"
20 PRINT "DEMONSTRIERT DEN GEBRAUCH"
30 PRINT "VON ZEILENNUMMERN"
40 LET A=2
50 LET B=4
60 LET C=A+B
70 PRINT C
80 END
```

Im praktischen Gebrauch ist der Unterschied zwischen direktem und indirektem Programmier-Modus eigentlich nur daran zu erkennen, daß im indirekten Modus die Befehle durch eine vorangestellte Zeilennummer einzugeben sind. Die Annahme, daß der direkte Modus dem RUN- und der indirekte Modus dem PRO-Modus entspricht, ist falsch. Denn auch im PRO-Modus, der in der Tat hauptsächlich zur Programmierung vorhanden ist, lassen sich einige Befehle direkt ausführen. Der Computer pendelt dabei automatisch zwischen direktem und indirektem Modus hin und her, wobei als Unterscheidungskriterium einzig und allein das Fehlen oder Vorhandensein einer Zeilennummer maßgebend ist.

Um ein Programm jedoch starten zu können, muß sich der Computer im RUN-Modus befinden, der ausschließlich zur direkten Befehlsverarbeitung vorgesehen ist. Die Umschaltung vom PRO- in den RUN-Modus erfolgt durch Betätigung der Taste [MODE]. Sobald der Computer bereit ist, eine neue Aufgabe zu übernehmen, erscheint auf dem Display das Bereitschaftszeichen > , der sogenannte "system prompt". Dieses Zeichen steht immer am Anfang der Zeile, in die die neuen Befehle eingegeben werden können.

8.2 Start und Ablauf eines Programmes

Programme sind nur dann ausführbar, wenn sich der Computer im RUN-Modus befindet. Ist diese Bedingung erfüllt, läßt sich das im Speicher befindliche Programm auf verschiedenste Art und Weise starten. Die wohl am häufigsten angewendete Methode ist dabei die der Eingabe des Kommandos RUN. Wird dieser Befehl mit Betätigung der Taste [ENTER] aktiviert, startet der Computer das Programm mit der niederwertigsten Zeile. Die weitere Programmausführung setzt sich dann Zeile für Zeile zu höherwertigen Zeilen hin fort.

Im vorangegangenen Programm-Beispiel wird also die Programmausführung mit Zeile 10 begonnen und mit den Zeilen 20, 30 usw. fortgesetzt, bis der in Zeile 80 stehende END-Befehl den Ablauf beendet. Wenn Sie diesen Ablauf in Realität verfolgen wollen, sollten Sie das betreffende Programm-Beispiel im PRO-Modus eintippen und anschließend im RUN-Modus mit dem Befehl RUN starten.

Beachten Sie, daß der Computer PC-1600 nach Beendigung einer Zeileneingabe durch Betätigung der [ENTER] Taste jeweils einen Doppelpunkt hinter die Zeilennummer setzt. Dieses soll Ihnen als Hinweis dienen, daß die eingegebene Zeile aus dem Eingabepuffer in den Arbeitsspeicher übernommen worden ist. Lassen Sie sich bitte auch nicht dadurch irritieren, daß der Doppelpunkt blinkt. Dieses ist lediglich ein Hinweis darauf, daß Sie, falls eine Korrektur der Zeile gewünscht ist, mit den Tasten [◄] und [►] an die zu ändernden Stellen gehen und die Berichtigungen vornehmen können.

Sind keine Korrekturen erforderlich, brauchen Sie nur die Nummer der nächsten Zeile eingeben, damit das Blinken stoppt und der Inhalt der neuen Zeile eingegeben werden kann. Nach Eingabe der letzten Zeile läßt sich das Blinken durch nochmalige Betätigung der [ENTER] Taste stoppen und die Programmeingabe beenden. Es erscheint dann auch wieder das Bereitschaftszeichen > auf dem Display.

Ein Programm muß nicht unbedingt mit der ersten Zeile in Betrieb genommen werden. Es läßt sich auch mit jeder anderen existenten Zeile starten. Hierfür stehen verschiedene Möglichkeiten bereit, die nachfolgend näher skizziert sein sollen. Der Grund für einen solchen ungewöhnlichen Programmstart mag bespielsweise sein, daß das Programm aufgrund eines Fehlers oder einer STOP-Anweisung abgebrochen wurde (s. Beschreibungen der Befehle CONT, GOTO und ARUN).

Programm-Marken und Gebrauch der [DEF]-Taste

Ein anderer und sehr flexibler Weg zum Start eines Programmes läßt sich mit Hilfe der [DEF]-Taste realisieren, die sich links neben den sechs Funktionstasten befindet. Mit dieser Taste läßt sich ein Programm ab einer speziell markierten Programmzeile starten. Als Markierungen sind hierbei ausschließlich folgende Großbuchstaben zulässig:

A , S , D , F , G , H , J , K , L ,
und Z , X , C , V , B , N , M .

Darüberhinaus können auch die Tasten [=] und [SPACE] als Lieferanten erlaubter Markierungen dienen.

Eine Programmzeile mit einer derartigen Markierung schaut wie folgt aus:

```
10:"A"
```

Beachten Sie, daß eine solche Marke stets in Anführungszeichen eingeschlossen werden muß, damit der Computer sie erkennen und beachten kann.

Ein Programm bzw. ein Programmteil, der mit einer derartigen Markierung versehen ist, läßt sich dadurch starten, daß man zuerst die Taste [DEF] betätigt und anschließend jene Taste, die das Markierungszeichen aufweist. Der Computer muß dabei im RUN-Modus betrieben werden.

Lassen Sie uns das bisher gesagte an unserem Beispielsprogramm einmal ausprobieren, indem wir die Zeile 10 entsprechend ändern:

```
10:"A":PRINT "DIESES EINFACHE PROGRAMM"
```

Denken Sie daran, daß für die Änderung der PRO-Modus eingestellt werden muß. Falls Sie mit den Editierfunktionen (s. Kapitel 4.5) noch nicht vertraut sein sollten, tippen Sie diese Zeile einfach komplett (so wie eben gezeigt) ein und drücken Sie anschließend die [ENTER]-Taste. Die zuvor im Speicher befindliche Zeile 10 wird so mit dem neuen Inhalt überschrieben. Stellen Sie danach wieder den RUN-Modus ein und starten Sie das Programm durch die Betätigung der Tastenkombination [DEF] + [A] .

Nach dieser Methode können also gezielt individuelle Programmteile gestartet werden oder aber unterschiedliche Programme, die gleichzeitig im Arbeitsspeicher stehen. Jeder Programmteil bzw. jedes unterschiedliche Programm braucht dazu nur mit einer anderen Markierung versehen sein. Solche Markierungen können aber auch dazu dienen, um im Zusammenhang mit speziellen BASIC-Befehlen bestimmte Zeilen zu adressieren ohne eine Zeilennummer angeben zu müssen. Beachten Sie hierzu bitte die Beschreibungen der Befehle LIST, DELETE und MERGE.

8.3 Speicherbelegung

Diejenigen Speicherbereiche des PC-1600, die dem Anwender für eigene Daten und Programme zur Verfügung stehen, können durch BASIC-Befehle aufgeteilt und gelöscht werden. Diese Befehle lauten:

CLEAR, ERASE, MEM, NEW, STATUS, TITLE

Schauen Sie hierzu auch im Anhang D (Speicher-Aufteilung) nach.

Zugängliche Speicherbereiche

Von dem 12 KByte (= 12288 Bytes) umfassenden eingebauten RAM des PC-1600 stehen dem Anwender 12090 Bytes für die Aufnahme eigener BASIC- und Maschinensprache-Programme zur Verfügung. Den Rest benötigt der PC-1600 für die Ablage von Daten und Zwischenergebnissen, die für den Betrieb des Systemes erforderlich sind. Darüberhinaus gibt es noch weitere Speicherbereiche, die für solche Systemzwecke vorbehalten sind und dem Computer-Benutzer normalerweise verborgen bleiben.

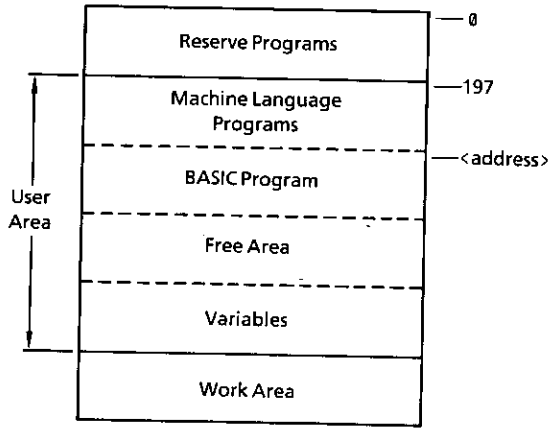


Abbildung 36: Aufteilung des Benutzer-Bereiches

1) Benutzer-Bereich (User Area)

Der Benutzer-Bereich stellt Speicherplatz für die Aufnahme der Variablen und der Programme zur Verfügung. Ohne Verwendung von zusätzlichen Modulen beträgt die Kapazität dieses Bereiches 12090 Bytes, d.h. ca. 12 KByte. Der Anschluß eines Diskettenlaufwerks (CE-1600F) reduziert diesen Bereich um 1064 Bytes.

Mit RAM-Modulen, die entweder im Modulfach S1 oder S2 eingesetzt werden, läßt sich diese Kapazität aufstocken und der Benutzerbereich damit bis in das Modul hinein ausdehnen. Dieses passiert auch dann, wenn ein Modul eingesetzt wird, von dem nur ein Teil als Speichererweiterung dient.

Um herauszufinden, wieviel Speicherplatz dem Benutzer insgesamt zur Verfügung steht, kann der MEM-Befehl eingegeben werden. Allerdings zeigt dieser Befehl nur den noch verbleibenden freien Platz des Benutzer-Bereiches an. Dieser freie Speicherplatz ist abhängig von der Größe des im Computer befindlichen Programmes. Haben Sie beispielsweise ein 4KByte-Modul in eines der Fächer S1 oder S2 eingesetzt, beträgt die Kapazität des insgesamt zur Verfügung stehenden Benutzerbereiches ca. 12 KByte + 4KByte. Laden Sie nun ein 3 KByte großes Programm, so verbleiben Ihnen im Anwenderbereich 16 KByte - 3 KByte = 13 KByte, die weiterhin frei genutzt werden können. Genau dieser Rest wird mit dem MEM-Befehl angezeigt.

2) Speicherbereich der Variablen (Variables Area)

Es ist nicht möglich, den wirklich frei bleibenden Speicherplatz vorauszusagen, wenn ein Programm im Anwender-Bereich geladen und noch nicht vollständig gelaufen ist. Diese Unberechenbarkeit liegt daran, daß das Programm mit Sicherheit während seines Ablaufes Variablen anlegt und diesen Werte zuweist. Der dafür erforderliche Speicherplatz wird vom Anwender-Bereich abgezogen. Der Betrag der erforderlichen Kapazität ist dabei abhängig von der Anzahl und Art der verwendeten Variablen. Wenn Sie jedoch diesen Speicherplatz löschen und dem Anwenderbereich wieder zur Verfügung stellen wollen, können Sie dieses mit Hilfe des CLEAR-Befehles tun.

3) Maschinensprache-Bereich (Machine Program Area)

Ein Teil des Benutzer-Bereiches kann auch von Maschinensprache-Programmen belegt werden. Diese Möglichkeit betrifft allerdings nur den fortgeschrittenen Anwender, da hierfür genaue Kenntnisse des Maschinen-Codes, also der Prozessorbefehle, und Erfahrungen im Umgang mit solchen Programmen erforderlich sind. Der für die Maschinensprache-Programme notwendige Speicherplatz kann mit dem Befehl NEW reserviert werden. Wird nach einem Reset der Befehl NEW 0 (s. Kapitel 3) eingegeben, wird der gesamte Benutzer-Bereich gelöscht und für die BASIC-Programmierung reserviert. Es steht dann kein Platz für die Aufnahme von Maschinensprache-Programmen zur Verfügung. Nach einer solchen Löschung sollte der MEM-Befehl einen freien Speicherplatz von 12090 Bytes anzeigen, sofern keine Module im Computer eingesetzt sind.

4) System-Bereich (Work Area)

Auf diesen Speicher-Bereich kann vom Benutzer nicht zugegriffen werden. Diesen Bereich benötigt der Computer zur Ausführung der einzelnen Prozeßschritte, die für die Abarbeitung von Programmen erforderlich sind.

5) Reserve-Bereich (Reserve Program Area)

Im Reserve-Bereich werden die Belegungen der programmierbaren Funktionstasten gespeichert, die im RESERVE-Modus diesen Tasten zugeordnet worden sind. Dieser Modus ist über die Bedienung der Tastenkombination SHIFT + MODE aktivierbar. In ihm lassen sich die Belegungen erstellen, ändern oder auf einem Speichermedium sichern bzw. von diesem laden. Der Reserve-Bereich hat eine feste Größe und unterliegt keiner dynamischen Variation.

9. Betriebsarten unter BASIC

9.1 Tastenfunktionen

Die Tastatur des Computers ist ähnlich aufgebaut wie bei einer Schreibmaschine oder anderen Computern. Dennoch gibt es einige Unterschiede. Mit speziellen Tasten oder Tastenkombinationen lassen sich im BASIC sehr nützliche Funktionen abrufen. Ebenso sind einige Tasten durch den BASIC-Interpreter in ihrer Wirkung beeinflußbar.

A) Aplhanumerische Tasten

Die Tasten des Alphabets, kurz Alphatasten genannt, sind genauso angeordnet, wie man es von einer Schreibmaschine her kennt. Sie können mit einer automatischen Wiederholfunktion belegt werden. Das bedeutet, daß jede Taste solange das ihr zugeordnete Zeichen erneut liefert, wie man die betreffende Taste gedrückt hält. Diese Wiederholfunktion läßt sich mit dem Befehl KEYSTAT ein- oder ausschalten. Mit diesem Befehl kann außerdem dafür gesorgt werden, daß jeder Tastendruck ein Klickgeräusch erzeugt, sofern dieses gewünscht ist. Die numerischen Tasten sind in einem Block zur rechten der Schreibmaschinen-Tastatur angeordnet.

B) Umschalt-Tasten

Mit den Umschalt-Tasten sind über die einzelnen Tasten mehrere Funktionen abrufbar, wobei mit den wohl am häufigsten benötigten Umschalttasten [SHIFT] über die Alphatasten sowohl die Groß- als auch die Kleinbuchstaben erzeugt werden können. Diese Umschalt-Tasten sind Ihnen bereits von einer Schreibmaschine her bekannt. Allerdings wirken sie dort in umgekehrter Weise, wie dieses nun beim PC-1600 der Fall ist.

[MODE] Mit dieser Taste kann zwischen den wichtigsten drei Betriebsarten umgeschaltet werden. Dabei ist bei einfacher Betätigung eine Hin- und Herschaltung zwischen dem RUN- und dem PRO-Modus möglich. Der RUN-Modus ist ausschließlich ein direkter Modus. In ihm werden alle eingegebenen BASIC-Befehle unmittelbar ausgeführt. Er ist darüberhinaus die einzige Betriebsart, in der sich BASIC-Programme starten lassen. Im PRO-Modus können BASIC-Programme erstellt, aufgelistet oder geändert werden. Der dritte Modus ist nur in Verbindung mit der [SHIFT] Taste aktivierbar.

[SHIFT] + [MODE] Zusätzlich zu den beiden gerade genannten Modi gibt es noch den sogenannten RESERVE-Modus. In diesem können die Belegungen der programmierbaren Funktionstasten erstellt, geändert oder angezeigt sowie geladen und gesichert werden. Um in diesen Modus zu gelangen, ist die Tastenkombination [SHIFT] + [MODE] zu betätigen. Dieser Modus kann durch einfache Bedienung der Taste [MODE] wieder verlassen werden.

[SHIFT] Wird die [SHIFT] -Taste gedrückt, so bleibt ihre Umschaltwirkung nur für die nachfolgend bediente Taste erhalten. Diese anschließend gedrückte Taste liefert dann ihre zweite Belegung, die auf dem Gehäuse in oranger Farbe symbolisiert ist. Verwechseln Sie den Begriff "zweite Belegung" bitte nicht mit der "zweiten Tastaturbelegung". Bei den Alphatasten wechselt [SHIFT] zwischen der Groß- und Kleinschrift der Buchstaben. Eine betätigte [SHIFT] -Taste bewirkt die Anzeige des gleichnamigen Symbols in der Statuszeile.

[CTRL] Diese Umschalttaste (control key) liefert in Verbindung mit den Tasten [↑] [↓] [←] [→] sowie den Tasten [A] [D] [E] [F] [H] [R] und [X] einige Editierfunktionen. Die Wirkung dieser Taste gilt nur für die unmittelbar anschließend betätigte Taste. Ist die [CTRL] -Taste gedrückt worden, so erscheint in der Statuszeile ein entsprechendes Symbol. Diese Anzeige erlischt zusammen mit der Wirkung dieser Taste, sobald man eine weitere Taste drückt.

[↕] Diese Menü-Taste schaltet zyklisch zwischen den drei Reserve-Ebenen um, wobei die Reihenfolge I, II, III gilt. Über jede dieser drei Ebenen kann auf eine andere Belegung der 6 Funktionstasten zugegriffen werden, so daß hiermit zwischen 18 verschiedenen Funktionen selektiert werden kann.

[SML] Diese Taste schaltet wechselseitig zwischen der Groß- und Kleinschrift der Alphatasten um. Ist diese Taste gedrückt worden und das gleichnamige Symbol in der Statuszeile sichtbar, so liefern die Alphatasten grundsätzlich Kleinbuchstaben und erst in Verbindung mit der Taste [SHIFT] die Großbuchstaben. In diesem Falle wirkt die Taste [SHIFT] dann genau wie bei einer Schreibmaschine.

DEF

Diese Taste (define key) kennt zwei Funktionen: 1) Im RUN-Modus lassen sich mit ihr Programme starten, die am Anfang mit einer Marke, die aus einem Buchstaben besteht (s.Kapitel 8), gekennzeichnet sind. 2) Mit ihrer Hilfe sind auch die standardmäßigen Belegungen der Funktionstasten abrufbar sowie die der obersten Alpha-Tasten.

RCL

Diese Taste (recall key) zeigt die Menüs der Funktionstasten F1 bis F6 an, die im RESERVE-Modus vereinbart worden sind. Diese Anzeige wird gelöscht, wenn die RCL-Taste nochmals betätigt wird. Das angezeigte Menü entspricht der derzeit gültigen Reserve-Ebene I,II oder III.

CLICK
KBII

Diese Taste schaltet das Klicken der Tasten an oder aus, wenn zuvor die SHIFT -Taste betätigt worden ist. Ohne vorherig gedrückte Umschaltung wird mit dieser Taste zwischen der ersten und zweiten Tastaturbelegung hin- und hergeschaltet. Ist die **zweite Tastaturbelegung** aktiv, erscheint in der Statuszeile das Symbol **S** und es können die internationalen Zeichen, die auf einer extra mitgelieferten Schablone vermerkt sind, von der Tastatur abgerufen werden.

C) Spezial-Tasten

Rechts vom numerischen Tastaturblock sowie unterhalb der Alpha-Tastatur befinden sich noch weitere Tasten, die die Bewegung des Cursors betreffen oder aber spezielle Bedeutung in Verbindung mit dem BASIC haben.

↓

Diese Taste hat drei Funktionen: 1) Im PRO-Modus wirkt ihre Betätigung wie ein LIST-Befehl, der die ersten Programmzeilen anzeigt. Mit weiteren Tastendrücken lassen sich auch die folgenden Programmzeilen schrittweise zur Anzeige bringen. 2) Tritt im Programm ein Fehler auf oder hält es wegen eines STOP- oder INPUT-Befehles an, so ist nach Umschaltung vom RUN- in den PRO-Modus durch diese Taste die betreffende Programmzeile lokalisierbar. 3) Ist der Trace-Modus aktiviert, so kann mit besagter Taste die nächste Programmzeile im PRO-Modus gestartet werden. Dieses gilt auch, wenn das Programm mittels STOP-Befehl oder Betätigung der BREAK-Taste gestoppt worden ist.

[↑] Diese Taste hat drei Funktionen: 1) Im PRO-Modus holt sie bei leerem Display die letzten Zeilen des Programmes in die Anzeige. Dann lassen sich mit ihr die vorangehenden Programmzeilen Zeile für Zeile sichtbar machen, bis der Anfang des Programmes erreicht ist. 2) Liegt aufgrund eines aufgetretenen Fehlers, eines STOP-Befehles oder aber der Betätigung der BREAK-Taste ein Abbruch eines Programmes vor, so läßt sich mit ihr die zuletzt bearbeitete Programmzeile in das Display holen. Dieses gilt auch für den Fall, daß der Computer, bedingt durch einen INPUT-Befehl, auf Eingaben wartet. 3) Ist der Trace-Modus aktiv, kann mit dieser Taste der Inhalt der gerade bearbeiteten Zeile angezeigt werden.

SHIFT + [↓] Diese Tastenkombination liefert im Anzeige-Modus MODE 1 das Zeichen π . Im MODE 0 wirkt sie wie wie die Taste [↓].

SHIFT + [↑] Diese Tastenkombination liefert im Anzeige-Modus MODE 1 das Zeichen $\sqrt{}$. Im MODE 0 wirkt sie wie wie die Taste [↑].

ENTER Diese Taste signalisiert dem Computer das Ende der momentanen Eingaben. Bevor diese Taste nicht gedrückt wird, bleiben alle Zeichen im Tastaturpuffer stehen, ohne daß sich der PC-1600 weiter um diese kümmert. Erst mit ihr wird er dazu aufgefordert, die im Puffer befindlichen Zeichen auszuwerten und zu interpretieren.

BS Mit dieser Taste kann das unmittelbar links vom Cursor befindliche Zeichen gelöscht werden. Alle rechts vom ihm stehenden Zeichen sowie das unter dem Cursor befindliche Zeichen rücken dann um eine Position nach links. Sofern sich der Cursor am physikalischen Zeilenanfang befindet, löscht er das von ihm verdeckte und bisher mitgeführte Zeichen. Erstreckt sich eine logische Zeile über mehrere Display-Zeilen, so geht der Cursor, am linken Rand der Display-Zeile angekommen, an das Ende der vorangehenden Display-Zeile und löscht das dort stehende Zeichen. Dieser Vorgang endet, wenn er den Anfang der logischen Zeile erreicht.

OFF Diese Taste schaltet den Computer aus.

ON Diese Taste schaltet den Computer an. Alle eingestellten Modi werden dann in der Statuszeile geeignet symbolisiert. Fortan wirkt diese Taste als Unterbrechungstaste [BREAK].

[BREAK] Mit dieser Taste läßt sich ein laufendes BASIC-Programm abbrechen. Auf dem Display erscheint dann die Meldung: BREAK IN LINE xxxx. Mit dem Befehl BREAK ON/OFF läßt sich diese Funktion an- bzw. abschalten.

[CL] Diese Taste löscht, sofern die Eingaben nicht mit der [ENTER] -Taste aktiviert worden sind, die komplette Eingabe-Zeile, in der sich der Cursor gerade befindet. Außerdem kann hiermit eine ERROR-Meldung gelöscht werden.

[CA] Diese über [SHIFT] + [CL] zugängliche Funktion löscht das Display und setzt interne Merker in den Bereitschaftsstatus.

[►] Die Taste mit dem nach rechts gerichteten Pfeil hat zwei Funktionen. 1) Im PRO-Modus bewegt sie den Cursor innerhalb einer Programmzeile um eine Position nach rechts. Befindet er sich am Ende einer logischen Zeile, bleibt diese Taste ohne Wirkung. 2) Im direkten Modus läßt sich mit ihr der letzte Rechenausdruck in die Anzeige zurückholen. Ergab sich bei dem Versuch seiner Berechnung ein Syntax-Fehler, so zeigt der blinkende Cursor dann auf die fehlerhafte Stelle.

[◄] Die Taste mit dem nach links gerichteten Pfeil hat ebenfalls zwei Funktionen. 1) Im PRO-Modus bewegt sie entsprechend den Cursor nach links, bis der Anfang der logischen Zeile erreicht ist. 2) Im direkten Modus wirkt diese Taste wie die mit dem nach rechts gerichteten Pfeil.

[DEL] Diese über die Kombination [SHIFT] +[◄] zugängliche Funktion löscht das vom Cursor verdeckte Zeichen und füllt die so entstandene Lücke dadurch auf, daß alle rechts vom Cursor befindlichen Zeichen um eine Position nach links verschoben werden.

[INS] Diese über die Kombination [SHIFT] +[►] zugängliche Funktion fügt an der momentanen Cursor-Position einen Platzhalter ␣ ein und schiebt alle rechts vom Cursor befindlichen Zeichen, einschließlich des unter dem Cursor befindlichen Zeichens, um eine Position nach rechts. Um mehrere Zeichen einfügen zu können, müssen durch mehrmalige Anwendung dieser Funktion genügend Platzhalter bereitgestellt werden, bevor diese mit den einzufügenden Zeichen aufgefüllt werden können.

9.2 Anzeige-Modi

Dieser Abschnitt beschreibt die zwei Anzeige-Modi des PC-1600, wie sie genutzt werden und erklärt das Koordinaten-System, mit dem Zeichen und Grafiken lokalisiert werden können.

Die zwei Anzeige-Modi werden MODE 0 und MODE 1 genannt, was Sie vielleicht verwirren und zu der Annahme verleiten mag, daß diese mit der MODE-Taste eingestellt werden können. Dem ist aber nicht so. Vielmehr wird zwischen diesen beiden Anzeigearten mit Hilfe des BASIC-Befehles MODE umgeschaltet, dessen genaue Erklärung in Kapitel 14 nachzulesen ist. Der standardmäßig angenommene Modus ist der MODE 0, der sich nach jedem Total-Reset des Computers automatisch einstellt.

MODUS 0 (26*4)

In diesem Modus wird das volle Display mit seinen 4 Zeilen zu je 26 Spalten für die Anzeige ausgenutzt. Jedes Zeichen wird dabei im Text-Modus durch eine 5x7-Punktmatrix gebildet.

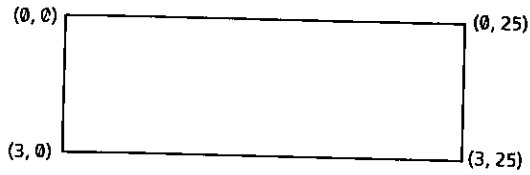


Abbildung 37 : Text-Koordinaten in MODE 0

Im Grafik-Modus (siehe dazu die Befehle: GCURSOR, GPRINT, LINE, POINT, PRESET und PSET) wirkt das gesamte Display als Matrix aus 156 * 32 Punkten, wobei jeder einzelne Punkt individuell gesetzt oder gelöscht werden kann. Hierzu dient zum Beispiel der Befehl PSET, dessen Parameter die X-Koordinate (0 bis 155) und die Y-Koordinate (0 bis 31) des Punktes bestimmen.

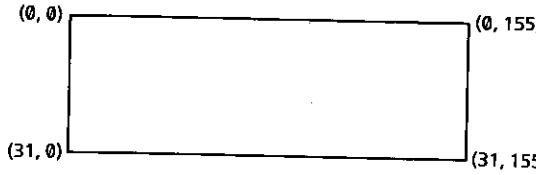


Abbildung 38 : Grafik-Koordinaten in MODE 0 und 1

MODUS 1 (26*1)

In diesem Anzeige-Modus wird von dem vierzeiligen Display lediglich eine Zeile für Darstellung von anzuzeigenden Daten genutzt. Dieser Modus erlaubt damit eine volle Kompatibilität zum Betrieb von PC-1500-Programmen. Die aktive Zeile ist stets die unterste. Die drei darüber befindlichen Zeilen bleiben in diesem Modus bei Anwendung der Befehle PRINT und GPRINT ungenutzt. Im PRO-Modus jedoch oder bei der Ausführung eines INPUT-Befehles rückt der Inhalt der untersten Zeile Zeile für Zeile nach oben.

Aus Kompatibilitätsgründen werden außerdem den ASCII-Codes &5B und &5D die PC-1500-üblichen Symbole √ und π zugeordnet.

Im Grafik-Modus wird, genau wie in MODE 0, das gesamte Display als Matrix von 156*32 Punkten angesprochen.

Abbildung 39 : Text-Koordinaten in MODE 1

Die Grafik-Befehle des BASICs erlauben die Darstellung einer Vielzahl von selbstdefinierten Symbolen, wobei die Befehle POINT und PSET die einzelnen Display-Punkte setzen. Insgesamt stehen folgende Grafik-Befehle zur Verfügung: GCURSOR, GPRINT, LINE, POINT, PRESET und PSET.

9.3 Editier-Modus

Eine mit Anweisungen gefüllte BASIC-Zeile kann bis zu 80 Zeichen lang sein. Das Display ist jedoch nur auf die Darstellung von 26 Zeichen pro Zeile ausgelegt. Daher werden längere BASIC-Zeilen in aufeinanderfolgenden Display-Zeilen angezeigt. Um eindeutig zwischen diesen beiden Zeilenarten unterscheiden zu können, verwenden wir oft die Begriffe "logische Zeile" und "physikalische Zeile". Eine logische Zeile meint eine komplette BASIC-Zeile, beginnend mit ihrer Zeilennummer bis zu ihrem inhaltlichen Ende von bis zu maximal 80 Zeichen. Eine physikalische Zeile meint immer eine Display-Zeile mit maximal 26 Zeichen Inhalt.

Die Editier-Funktionen des BASICs unterscheiden nicht zwischen physikalischen Zeilen, sondern beziehen sich immer auf logische Zeilen.

Programm-Zeilen können nur im PRO-Modus editiert werden, wobei verschiedene Methoden anwendbar sind. Die simpelste, aber oftmals aufwendigste, ist die, eine Zeile einfach komplett in neuer Form einzugeben und damit den bisherigen Zeileninhalt zu überschreiben. Alte und neue Zeile müssen demzufolge mit derselben Zeilennummer versehen sein. Mit Hilfe des LIST-Befehles kann die zu ändernde Zeile jedoch leicht in die Anzeige geholt und mit den Editier-Funktionen bearbeitet werden. Nach Abschluß dieser Arbeiten ist die [ENTER] Taste zu drücken, damit alle Änderungen übernommen werden und den alten Zeileninhalt überschreiben. Ohne Betätigung dieser Taste bleiben die Änderungen wirkungslos.

Ist ein Programm durch einen STOP-Befehl oder der Betätigung der BREAK-Taste oder aber aufgrund eines aufgetretenen Fehlers abgebrochen worden oder wartet der PC-1600 wegen eines INPUT-Befehls auf Eingaben, so kann mit der MODE Taste der PRO-Modus aktiviert werden. Eine Betätigung von [↑] oder [↓] zeigt dann die zuletzt ausgeführte Programmzeile an, wobei der Cursor so positioniert wird, daß mit dem Editieren begonnen werden kann. Man kann sich aber auch durch unmittelbar anschließende Betätigung der Taste [ENTER] die weiteren Programmzeilen anzeigen lassen.

Die [CTRL] Taste erlaubt im Zusammenhang mit weiteren Tasten ein einfaches Editieren von BASIC-Programmen. Dabei kann gezielt zu bestimmten Zeilen gesprungen werden, ohne sich umständlich Zeile für Zeile bis zum Ziel vortasten zu müssen. Stehen mehrere Programme im Arbeitsspeicher, kann mit dem GOTO-Befehl und Angabe einer Zeilennummer oder einer Marke das zu editierende Programm selektiert werden. Alle nachfolgenden LIST-, RENUM- und DELETE-Befehle beziehen sich dann ausschließlich auf dieses Programm.

[CTRL] + [↑] Die Betätigung dieser Tastenkombination zeigt auf dem Display die erste Programmzeile und bis drei weitere Programmzeilen an. Dabei ist der Cursor unsichtbar auf dem Doppelpunkt hinter der Zeilennummer positioniert. Zur Kennzeichnung seiner Position blinkt der Doppelpunkt.

[CTRL] + [↓] Die Betätigung dieser Tastenkombination zeigt auf dem Display die letzte Programmzeile sowie bis zu drei vorangehende Programmzeilen an. Der Cursor wird dabei unsichtbar auf dem blinkenden Doppelpunkt positioniert.

[CTRL] + [→] Diese Tastenkombination setzt den Cursor an das Ende der logischen Zeile, sofern er nicht auf dem blinkenden Doppelpunkt positioniert ist.

[CTRL] + [←] Diese Tastenkombination setzt den Cursor an den Anfang der logischen Zeile (also auf die erste Ziffer der Zeilennummer), sofern er sich nicht auf dem blinkenden Doppelpunkt befindet.

CTRL + A	Diese Taştenkombination schaltet zwischen dem Einfüge- (Insert) und dem Überschreibe-Modus (Overwrite) hin- und her. Welcher Modus gerade aktiv ist, kann anhand des blinkenden Cursors oder Doppelpunktes erkannt werden. Im Einfüge-Modus ist die Blinkfrequenz höher.
CTRL + D	Mit dieser Tastenkombination lassen sich alle Zeichen, die links vom Cursor stehen, löschen. Diese Löschung schließt auch die Zeilennummer mit ein.
CTRL + E	Mit dieser Tastenkombination lassen sich alle Zeichen, beginnend bei der Cursor-Position, bis zum Ende der logischen Zeile hin löschen.
CTRL + F	Diese Tastenkombination setzt den Cursor auf das erste Zeichen des nächsten in der Zeile befindlichen Wortes.
CTRL + H	Diese Tastenkombination löscht das unmittelbar links vom Cursor stehende Zeichen und rückt den Rest der Zeile um eine Position nach links. Ist der Cursor am Anfang der logischen Zeile, so wird das an dieser Position befindliche Zeichen gelöscht und der Rest der Zeile nach links ververschoben. Diese Tastenkombination hat damit dieselbe Wirkung wie die BS -Taste.
CTRL + R	Diese Tastenkombination schaltet die Wiederholfunktion der Tasten ein oder aus.
CTRL + X	Diese Tastenkombination löscht das Display und hat damit dieselbe Wirkung wie die CL -Taste.

9.4 Reserve-Modus

Im RESERVE-Modus erhalten Sie den Zugriff auf einen speziellen Speicherbereich, in dem die Belegungen der sechs Funktionstasten abgelegt sind. Diese Funktionstasten befinden sich direkt unterhalb des Display-Fensters und sind mit den Zeichen ! " # $ % und & markiert.

Die Zeichenkette (string), die eine dieser Funktionstaste auf Betätigung hin liefert, kann vom Anwender nach Wunsch geändert werden. Diese Strings lassen sich auch auf einem Speichermedium sichern und von diesem laden, so daß der Anwender jederzeit die gewünschte Belegung der Funktionstasten nach seinen Bedürfnissen gestalten kann.

A) Auswahl der RESERVE-Modi

Der RESERVE-Modus läßt sich durch die Kombination der beiden Tasten [SHIFT] + [MODE] aktivieren. In der Statuszeile wird dann der Schriftzug RESERVE angezeigt. Durch einfache Betätigung der Taste [MODE] kann man diesen Modus wieder verlassen.

Im RESERVE-Modus lassen sich drei Belegungsebenen unterscheiden, die in der Statuszeile durch die Symbole I, II und III kenntlich gemacht sind. Mit der sogenannten Menü-Taste [▲▼] kann zwischen diesen drei Ebenen zyklisch umgeschaltet werden. In jeder Ebene können jeweils sechs verschiedene Strings den programmierbaren Funktionstasten zugeordnet werden, so daß dem Anwender insgesamt 18 unterschiedliche Belegungen zur Verfügung stehen.

B) Setzen der Funktionstasten-Strings

1) Wählen Sie zuerst mit der Tastenkombination [SHIFT] + [MODE] den RESERVE-Modus an. Geben Sie dann über die Tastatur den Befehl NEW ein und betätigen Sie die [ENTER] -Taste. Damit werden alle existierenden Belegungen und Menüs gelöscht. Falls irgendeine dieser Belegungen erhalten bleiben soll, ist dieser Schritt folglich zu vermeiden.

2) Wählen Sie mit der Menü-Taste die Ebene I aus, um damit die Funktionstasten F1 bis F6 mit einem ersten Satz an Belegungen versehen zu können.

3) Drücken Sie Funktionstaste F1 und geben Sie die Zeichenkette, die Sie als Belegung wünschen, ein. Dabei sind alle Zeichen des Alphabets, alle numerischen Zeichen und Symbole erlaubt. Schließen Sie diese Eingabe durch Betätigung der [ENTER] -Taste ab.

4) Wiederholen Sie Schritt 3) für die Funktionstasten F2 bis F6.

5) Wählen Sie dann Ebene II durch Druck auf die Menü-Taste [▲▼].

6) Wiederholen Sie die Schritte 3) und 4), um die programmierbaren Tasten mit einer zweiten Belegung zu versehen.

7) Wählen Sie durch Betätigung der Taste [▲▼] die Ebene III an und wiederholen Sie nun die Schritte 3) und 4) für eine dritte und letzte Belegung der sechs Funktionstasten.

Nun sollten den Funktionstasten 18 Strings zugewiesen sein, die durch einfachen Druck auf diese Tasten abrufbar sind.

Die möglichen Belegungen können BASIC-Befehle, Funktionsnamen oder aber auch Rechenausdrücke sein. Sollen die Befehle direkt zur Ausführung gelangen, müssen sie mit einem Klammeraffen @ abgeschlossen sein.

Ein Beispiel erlaubter und sinnvoller Belegungen wäre:

Taste	Ebene I	Ebene II	Ebene III
F1	SIN	FOR	PRINT
F2	COS	TO	LIST
F3	TAN	STEP	AUTO
F4	LOG	NEXT	LOAD
F5	EXP	GOTO	SAVE
F6	INT	RUN	LLIST

Die Länge der zuweisbaren Strings wird dadurch bestimmt, daß maximal 110 Bytes pro Funktionstaste zugewiesen werden können. Ein BASIC-Befehl oder eine Funktion (z.B. RUN, PRINT, SIN, COS) benötigen jeweils zwei Bytes an Speicherplatz, da sie intern in sognannte Tokens umgewandelt werden. Jedes andere Zeichen, das nicht zu einem BASIC-Wort gehört, belegt jeweils ein Byte. Eine Funktionstaste kann also mehr als nur einen Befehl aufnehmen, sondern auch ganze Anweisungen, wie z.B.:

F1 FOR I=0 TO 100 STEP 2

Sie sollten Ihre Belegungen jedoch so wählen, daß sie die von Ihnen am häufigsten gebrauchten Befehlswörter oder Anweisungen enthalten, um so die Arbeit des Programmierens zu beschleunigen und angenehmer zu gestalten.

C) Abruf der Funktionstasten-Belegungen

Die den Funktionstasten im RESERVE-Modus zugewiesenen Strings können im RUN-Modus während der Eingaben direkter Berechnungen oder im PRO-Modus während des Programmierens abgerufen werden. Dazu braucht mit der Menü-Taste lediglich die gewünschte Ebene (I,II oder III) selektiert und die passende Funktionstaste gedrückt zu werden. Der String wird dann an der Cursor-Position im Display ausgegeben. In unserer Beispiel-Belegung würde die Betätigung der Funktionstaste F1 bei der ausgewählten Ebene III den String PRINT liefern.

Der Speicherbereich, der die Funktionstastenbelegungen der Ebene II aufnimmt, wird auch für die Speicherung der Alarmmeldung verwendet. Sofern über ALARM$ eine solche Alarmmeldung vereinbart wird, werden alle Funktionstastenbelegungen dieser Ebene durch diesen Alarm-String überschrieben.

D) Funktionstasten-Menüs

Da die Belegungen der Funktionstasten erst bei deren Betätigung sichtbar werden, kann es vorkommen, daß man sich nicht aller 18 Belegungen gegenwärtig ist, und weiß, welche Taste zu drücken ist, um eine bestimmte Belegung abzurufen. Deshalb besteht die Möglichkeit, durch Betätigung der [RCL] -Taste (recall), eine als Menü bezeichnete Orientierungshilfe im Display darstellen zu lassen. Diese kann wegen der Beschränkung auf 26 anzeigbare Zeichen natürlich nur eine verkürzte Form der wirklichen Belegungen wiedergeben.

Erzeugung eines Menüs:

1) Setzen Sie den Computer in den RESERVE-Modus durch Betätigung der Tastenkombination [SHIFT] + [MODE] .

2) Wählen Sie mit der Menü-Taste die Belegungs-Ebene für die Sie ein Menü erstellen möchten.

3) Geben Sie nun einen Menü-String aus bis zu 26 Zeichen ein, der in Anführungsstrichen eingeschlossen sein muß. Beenden Sie die Eingabe mit Betätigung der [ENTER] -Taste.

Dieses Menü wird dann in einem Teil des Reserve-Speichers für einen späteren Abruf bereitgehalten. Denken Sie daran, daß die maximale Länge des Menü-Strings genau 26 Zeichen beträgt, also der Länge einer Display-Zeile. Damit das Menü so gestaltet ist, daß die Kürzel über den entsprechenden Tasten liegen, sollte jedes Kürzel nicht länger als 4 Zeichen sein.

Für unsere zuvor genannte beispielhafte Belegung könnte das Menü für die dritte Belegungsebene folgendermaßen gewählt werden:

```
PRNT LST AUT  LD SVE LLST
<-------26 Zeichen------->
```

Anzeige eines Menüs

Das Beispiel-Menü kann im PRO- oder RUN-Modus angezeigt werden, indem man die Belegungsebene III mittels Menü-Taste anwählt und die [RCL] Taste betätigt. Eine abermalige Betätigung der [RCL] Taste löscht die Anzeige des Menüs. Auf ähnliche Weise lassen sich die Menüs der Belegungsebenen I und II zur Anzeige bringen.

Das Beispiel-Menü für die Ebene III erscheint auf dem Display im folgenden Verhältnis zu den Funktionstasten:

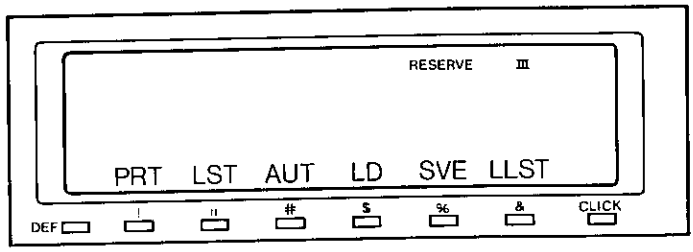


Abbildung 40 : Beispiel einer Menü-Anzeige

E) Speichern und Laden von Funktionstasten-Belegungen

Die den Funktionstasten zugeordneten Strings lassen sich genau wie BASIC-Programme speichern oder laden. Der Computer muß sich dazu allerdings im RESERVE-Modus befinden bevor die bekannten Befehle SAVE und LOAD eingegeben werden.

ACHTUNG: Wird eine Tastaturbelegung im PRO-Modus geladen, hat dieses zur Folge, daß die im Arbeitsspeicher befindlichen Programme zerstört werden. Umgekehrt gilt das Entsprechende, wenn man im RESERVE-Modus anstelle von Tastaturbelegungen Programme lädt.

F) Vordefinierte Tastaturbelegung

Eine Reihe der häufigst gebrauchten BASIC-Befehle sind bereits den in der oberen Tastaturreihe befindlichen Aplhatasten fest zugeordnet. Auf die Befehlswörter, die diesen Tasten zugewiesen sind, läßt sich mit Hilfe der DEF Taste zugreifen, wobei die folgenden Belegungen gelten:

Taste	Befehlswort
Q	INPUT
W	PRINT
E	USING
R	GOTO
T	GOSUB
Y	RETURN
U	CSAVE
I	CLOAD
O	MERGE
P	LIST

Die Befehle CSAVE, CLOAD, MERGE und LIST können dabei nur dann benutzt werden, wenn ein Cassetten-Recorder über den Drucker CE-1600P an den Computer angeschlossen ist.

Die genannten Befehlswörter sind auf einer der mitgelieferten Tastaturschablonen eingraviert. Legen Sie diese Schablone über die Tastatur, damit eine schnelle Referenz gewährleistet ist.

Zusätzlich können mit der [DEF] Taste sechs weitere Befehle abgerufen werden, die sozusagen als vierte Belegung den Funktionstasten zugeordnet sind. Diese lauten:

[DEF] + [!]	RUN	ENTER
[DEF] + ["]	AUTO	
[DEF] + [#]	LOAD"	
[DEF] + [$]	SAVE"	
[DEF] + [%]	FILES"	
[DEF] + [&]	"COM1:"	

10. Datendarstellung

Der Taschencomputer PC-1600 basiert auf digitalen Bauelementen, die im Grunde nur die Informationen "ja" oder "nein" bzw. die binären Werte "1" und "0" voneinander unterscheiden können. Eine solche logische Unterscheidung trägt die Informationseinheit 1 Bit. Alle Daten, egal welcher Art sie auch sein mögen, müssen intern also auf diese binäre Darstellung zurückgeführt, d.h. in eine Vielzahl geeigneter Bits zerlegt werden, damit der Computer sie verarbeiten kann. Da der PC-1600 mit einem 8-Bit-Prozessor bestückt ist, kann jeweils eine Einheit von 8 Bits gleichzeitig zur Verarbeitung herangezogen werden. Eine solche Einheit aus 8 Bits nennt man auch ein Byte. Mit einem solchen Byte können alle ganzen Zahlen von 0 bis 255 dargestellt werden. Die Zahl Null wäre hierbei durch das Bitmuster 00000000, die Zahl 17 durch das Muster 00010001 und die Zahl 255 durch 11111111 repräsentiert. Diese Darstellungsform erweist sich bei Auflistung in Tabellen als unübersichtlich und recht anfällig für Ablesefehler. Deshalb hat man sich für die Darstellung eines Bytes etwas anderes ausgedacht, die sogenannte hexadezimale Notation. Sie ist kürzer und beschreibt ein Byte mit genau zwei Hexadezimalziffern, wobei jede dieser Ziffern damit 4 Bits ersetzt. Da sich mit 4 Bits genau 16 verschiedene Zustände beschreiben lassen, muß dieses mit einer Hexadezimalziffer ebenfalls so sein. Alle Hexadezimalzahlen müssen folglich mit einem Vorrat aus 16 Ziffern gebildet werden können. Bei dem uns vertrauten dezimalen Zahlensystem steht ein Vorrat von zehn Ziffern (0 bis 9) zur Verfügung. Um nun für die Hexadezimalziffern nicht neue Symbole entwerfen zu müssen, hat man vereinbart, daß die ersten zehn hexadezimalen Ziffern die Symbole der Dezimalziffern tragen sollen und die verbleibenden sechs Hexadezimalziffern durch die ersten sechs Großbuchstaben des lateinischen Alphabets vertreten sein sollen. Somit lauten die Hexadezimalziffern:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E und F.

Damit Dezimal- und Hexadezimalzahlen eindeutig voneinander zu unterscheiden sind, ist es üblich, die Hexadezimalzahlen zu kennzeichnen. Wir werden es im folgenden dadurch tun, daß wir der Hexadezimalzahl unmittelbar das Zeichen & (Kaufmannsund) voranstellen. Diese Kennzeichnung ist auch zu verwenden, wenn in den PC-1600 hexadezimale Werte eingegeben werden.

Geben Sie beispielsweise die Ziffernfolge 27 in den Rechner ein, entspricht dies offensichtlich dem Wert siebenundzwanzig. Tippen Sie aber die Zeichenfolge &27 (sprich: hex zwei sieben) ein, entspricht dieses jedoch dem Wert neununddreißig.

Die Werte eines Bytes lassen sich also mit den Hexadezimalzahlen von &00 bis &FF darstellen.

10.1 Datentypen

A) Textdaten und Zeichensätze

Wie bereits erwähnt, lassen sich mit einem Byte die Zahlenwerte 0 bis 255 darstellen. Um auch größere Werte, gebrochene oder gar negative Zahlen verarbeiten zu können, müssen diese in mehrere Bytes aufgeteilt werden. Darüberhinaus möchte man aber mit dem Computer auch Daten verarbeiten können, die keinen Zahlenwert tragen. Dieses können Buchstaben, Satzzeichen, Symbole oder aber Steuerzeichen sein. Unter Anwendung eines Bytes lassen sich 256 solcher Zeichen erfassen. Damit zwischen einem Computer und dem daran angeschlossenen Gerät eine erfolgreiche Kommunikation zustande kommen kann, müssen diese Geräte unter jedem der 256 möglichen Bytewerte jeweils dasselbe Zeichen verstehen. Es muß also eine genormte Vereinbarung über die Zuordnung zwischen Bytewert und Zeichen bestehen. Eine solche Vereinbarung nennt man Codierung oder schlicht einen Code. Weltweit sind besonders die drei Codes ASCII (USA), EBCDIC (Europa) und JIS (Japan) verbreitet, wobei sich allerdings bei kleineren Computern der ASCII-Code als weltweiter Standard behauptet hat.

Arbeiten also zwei kommunizierende Geräte gemeinsam nach einem dieser drei Codes, so sollte man eigentlich eine 100-prozentige Verständigung zwischen beiden erwarten können. Aber wie sooft wenn Regeln oder Gesetze aufgestellt werden, gibt es jemanden, der aus irgendeinem Grunde sich nicht hieran hält und diese nach eigenem Geschmack interpretiert und ändert. Leider gilt dieses auch bei den erwähnten Standard-Codes, insbesondere aber für den ASCII-Code. Je nach Hersteller können Unterschiede bei den Codes der grafischen Symbole auftreten. Dieses ist auch nicht weiter verwunderlich, da der ASCII-Code eigentlich nur ein 7-Bit-Code ist und damit nur für die Werte 0 bis 127 (&0 bis &7F) genormt wurde. Desweiteren existieren nationale Abwandlungen, die die landesspezifischen Sonderzeichen berücksichtigen. In Deutschland sind dieses die Zeichen: Ä, Ö, Ü, ä, ö, ü, ß und § . Dadurch verzichtet man aber auf die eckigen und geschweiften Klammern. Um diesem Problem aus dem Wege zu gehen, belassen heutzutage viele Hersteller den ursprünglichen ASCII-Code für die Werte &0 bis &7F unverändert und legen im Bereich der 8-Bit-Erweiterung (&80 bis &FF) die landesspezifischen Symbole mit zu den Grafik-Symbolen. Der PC-1600 richtet sich hierbei nach der Codierung des IBM-PC.

Der PC-1600 und sein Vorgänger PC-1500 weisen unterschiedliche Zeichensätze auf. Dieses berücksichtigt der PC-1600 bei der Umschaltung von Anzeigemodus MODE 0 in den PC-1500-kompatiblen MODE 1 und umgekehrt, indem er die abweichenden Zeichen den betreffenden Codes richtig zuordnet. Es handelt sich hierbei um die Zeichen mit den Codes &5B, &5D und &27, die beim PC-1500 folgende Bedeutung haben:

&27	Einfüge-Symbol	⌴
&5B	Wurzelzeichen	√
&5D	Symbol Pi	π

B) Zeichen-Strings

Aus den Text-Zeichen können Ketten mit einer Länge von bis zu 80 Zeichen gebildet und vom Computer verarbeitet werden. Solche Zeichenketten nennt man in der Computertechnik Strings. Diese besitzen keinen numerischen Wert. Mit ihnen kann also weder gerechnet werden, noch sind sie numerischen Variablen zuweisbar. Aus diesem Grunde muß der Anwender numerische Daten und Strings streng voneinander unterscheiden. Diese Unterscheidung bedingt in der Programmiersprache BASIC einer speziellen Notation.

C) Numerische Daten

Alle numerischen Daten werden intern in ein binäres Format umgewandelt und in dieser Gestalt gespeichert. Die Eingaben können jedoch in dezimaler, in wissenschaftlicher oder in hexadezimaler Schreibweise erfolgen. Bei einer Berechnung werden die einzelnen binären Daten miteinander verknüpft und das binäre Resultat so gewandelt, daß es in dezimaler Form auf dem Display angezeigt werden kann. Werden die Werte zu groß oder zu klein, um sie mit einer begrenzten Ziffernanzahl darstellen zu können, erfolgt die Anzeige automatisch in wissenschaftlicher Notation, d.h. mit Exponenten-Angabe. Unter gewissen Bedingungen kann mit Hilfe des Befehles HEX$ die Ausgabe auch in hexadezimaler Form erfolgen.

Intern benutzt der Computer eine binäre Fließkomma-Darstellung für die Erfassung der numerischen Daten. Diese Darstellung ist so ausgelegt, daß jeder dezimale Wert durch eine 12-stellige Mantisse und einen 2-stelligen Exponenten repräsentiert wird. Die Mantisse wird bei der Ausgabe auf 10 signifikante Stellen gerundet. Die numerischen Ein- und Ausgaben können damit im Bereich $\pm$ ($9.999999999 \times 10^{-99}$ ... $9.999999999 \times 10^{99}$) liegen. Durch die besagte Rundung kann die letzte angezeigte Ziffer der Mantisse um den Wert $\pm$ 1 fehlerhaft sein.

10.2 Konstanten

A) String-Konstante

Eine String-Konstante ist eine Zeichenkette, die in Anführungszeichen eingeschlossen ist. Beispiele sind:

```
"SHARP PC-1600"
"ABCDEF"
"Die Taschencomputer-Revolution"
"1.256 bis 84.4"
```

Der Computer unterscheidet nicht zwischen numerischen Zeichen, Alphazeichen oder anderen Symbolen, solange diese in Anführungszeichen eingeschlossen sind. Ein Paar direkt aneinandergefügter Anführungszeichen schließt keine Zeichen ein und repräsentiert damit einen sogenannten Leer- oder Null-String, der intern durch den ASCII-Code 0 vertreten wird. Die Anwendung eines solchen Strings kann bei der Programmierung oftmals von großem Nutzen sein (siehe INKEY$).

Die maximale Länge einer Stringkonstanten beträgt 80 Zeichen.

B) Numerische Konstante

Numerische Konstante können positive oder negative Zahlen sein, die man in vier Typen unterscheidet:

a) Integer-Konstante (Ganzzahl-Konstante)

Bei diesen Konstanten handelt es sich um ganze Zahlen. Beim PC-1600 können diese im Wertebereich -32768....+32767 liegen. Negativen Werten wird ein Minuszeichen vorangestellt. Fehlt ein Vorzeichen, wird der positive Wert angenommen.

Beispiele: 123, +2, -57, 1024

b) Fixkomma-Konstante

Diese Konstanten weisen im Gegensatz zu den Integer-Werten einen Dezimalpunkt (Komma) auf, besitzen also Stellen vor und hinter dem Komma. Auch hier gilt ein fehlendes Vorzeichen als ein Pluszeichen.

Beispiele: 12.7, +1.345, 0.222232, -67.9888

c) Fließkomma-Konstante

Diese Konstanten weisen außer den Ziffern einer Integer- oder Fixkomma-Konstanten noch einen Exponenten auf. Dieser wird in der Anzeige durch den Buchstaben E kenntlich gemacht. Sowohl die Mantisse als auch der Exponent können vorzeichenbehaftet sein. Kein Vorzeichen bedeutet auch hier Identität mit einem Pluszeichen.

d) Hexadezimale Konstante

Diese Konstanten werden nicht im dezimalen, sondern im hexadezimalen Zahlensystem repräsentiert und deswegen mit einem vorangestellten Kaufmannsund (&) charakterisiert.

Beispiele: &23, &8000, &2B, &FFFF

Hexadezimale Konstanten dürfen im Bereich &0 (Null) bis &FFFF (= 65535 dezimal) liegen. Negative oder gebrochene Zahlen sind nicht erlaubt.

10.3 Variablen

Variablen sind Bezeichner, die temporär als Stellvertreter für bestimmte Werte dienen können. Ihnen lassen sich jederzeit neue Werte zuweisen. Bei der Verwendung solcher Variablen werden im Computer ihre Namen gespeichert und Platz für die Aufnahme der aktuellen Werte reserviert. Sowohl die Namensgebung als auch die Belegung der Variablen mit Werten obliegt dem Anwender. Solche Zuweisungen von Werten können auch per Programm geschehen.

Da zwei grundlegende Datentypen zu unterscheiden sind, nämlich numerische Daten und Strings, gibt es folglich auch zwei unterschiedliche Variablentypen. Numerische Variablen dienen der Aufnahme von numerischen Daten und Stringvariablen der Aufnahme von Strings. Numerische Variablen finden Verwendung zur Speicherung von Zahlen und zu deren Repräsentation in Berechnungen. Ihre Werte werden intern im Fließkomma-Format bearbeitet und entweder in der Integer- oder Fließkomma-Darstellung ausgegeben.

Beispiel: A=12.6043 VA=1.2345668E-3

String-Variablen dienen der Speicherung von zusammengeketteten ASCII-Zeichen.

Außer der Vereinbarung eines Namens muß bei einigen Variablen auch noch eine gesonderte Speicherplatzreservierung vorgenommen werden. Dieses wird durch Anwendung des Befehles DIM möglich. Eine numerische Variable hat den Wert Null, solange ihr kein Wert zugewiesen worden ist, eine Stringvariable ist dagegen in solchem Falle leer.

A) Variablen-Namen

Namen numerischer Variablen können aus beliebig vielen Zeichen bestehen, wobei jedoch nur die ersten zwei Zeichen signifikant sind und vom Computer beachtet werden. Damit sind für ihn die Variablen-Namen TOTAL und TOP identisch. Für die Namensgebung gilt außerdem, daß das erste Zeichen immer ein Großbuchstabe von A bis Z sein muß. Alle weiteren Zeichen können ebenfalls Großbuchstaben oder aber Ziffern sein.

Alle Bezeichnungen, die einem BASIC-Befehlswort entsprechen oder ein solches in sich aufweisen, sind nicht erlaubt.

Beispiel nicht erlaubter Bezeichnungen:

APFELSINE enthält SIN usw....

Für die Benennung von Stringvariablen gelten dieselben Regeln wie für numerische Variablen. Zusätzlich müssen sie jedoch mit einem Dollarzeichen ($) gekennzeichnet sein, das am Ende des Namens erscheinen muß.

Damit sind gleichlautende numerische Variable und String-Variable zulässig, wie z.B.: AB und AB$.

Sowohl numerische als auch Stringvariablen können darüberhinaus mit Indizes versehen werden, wobei der Index in runden Klammern dem Variablen-Namen nachgestellt sein muß. Der Index muß ein ganzzahliger Wert sein, der entweder direkt angegeben ist oder aber durch eine numerische Variable vertreten wird. Ebenso kann er durch einen numerischen Ausdruck gebildet sein.

Beispiele gültiger Variablen-Bezeichnungen:

numerische Variablen:	A, C1, D9, RE(6), X2(I)
String-Variablen:	CC$, B$, D4$, NO$(4), TR$(A)

B) Variablen-Typen

Außer in numerische und String-Variablen unterscheidet das BASIC noch weitere drei übergeordnete Typisierungen von Variablen:

a) die Standard-Variable,
b) die einfache Variablen und
c) die Array-Variablen (oder einfach: Arrays)

Variablen-Typ	numerisch	String
Standard	A .. Z	A$ .. Z$
einfach	A0 .. Z9 AA .. ZZ	A0$ .. Z9$, AA$ .. ZZ$
Array	A0(0) .. Z9(255) AA(0) .. ZZ(255)	A0$(0) .. Z9$(255) AA$(0) .. ZZ$(255)

a) Numerische Standard-Variablen

Die Standard-Variablen sind eine Besonderheit des SHARP-BASICs. Für sie wird vom Computer ein fester Speicherbereich reserviert, der weder durch andere Variablen belegt noch für andere Zwecke verwendet werden kann (siehe Anhang D). Der Speicherplatz hat eine feste Größe und ist unabhängig von der Belegung der darin gespeicherten Variablen. Die Standardvariablen können auch als ein gemeinsames eindimensionales Array angesprochen werden, wobei das Symbol @ (Klammeraffe) als Variablen-Name dient. So greift beispielsweise @(1) auf den Speicherplatz A und @(26) auf den Speicherplatz Z zu.

Die Inhalte der Standard-Variablen können mit dem Befehl CLEAR gelöscht werden. Einfache Variablen und Arrays werden dabei mitgelöscht. Um nur einfache und Array-Variablen zu löschen und Standard-Variablen unberührt zu lassen, sollte der Befehl ERASE verwendet werden.

b) Einfache numerische Variablen

Einfache numerische Variablen werden in einem Speicher-Bereich oberhalb des BASIC-Programmes abgelegt. Dieser Bereich ist in seiner Größe davon abhängig, wieviele Variablen generiert und mit Werten belegt werden, welcher Art diese Werte und welchen Typs sie sind. Die Größe dieses Bereiches ist also dynamisch veränderlich. Je mehr Variablen innerhalb des Programmes benutzt werden, umso weniger Platz bleibt für das Programm selbst übrig. Liegt ein sehr umfangreiches Programm vor, so kann es im Extremfall dazu kommen, daß kein Platz mehr für die Generierung und Ablage von weiteren Variablen zur Verfügung steht. Dann muß die Speicherkapazität des Computers durch geeignete RAM-Module erweitert werden.

c) Numerische Felder (Arrays)

Ein numerisches Feld, auch Array oder Matrix genannt, stellt eine Gruppe gleichnamiger numerischer Variablen dar, die nur durch einen anderen Index voneinander unterschieden werden. Dieser Index muß vom Typ Integer, also eine ganze Zahl sein.

Beispiel: A(2) ist das dritte Element des Arrays A(X)
A(11) ist das zwölfte Element des Arrays A(X)

Ein Array bzw. Feld kann eindimensional oder zweidimensional sein. Bei zweidimensionalen Feldern folgen dem Variablen-Namen zwei Indizes, die durch ein Komma voneinander zu trennen sind:

X(3,4) , AB(4,12) usw.

Eine Array-Variable kann immer nur entweder ein- oder zweidimensional sein. Deshalb sind die Variablen BM(3) und BM(2,5) gleichzeitig nicht im Speicher des Computers realisierbar.

Ein eindimensionales Array bezeichnet man auch als Liste, ein zweidimensionales Array dagegen als Tabelle. Folglich ist ein Array eine Liste oder eine Tabelle von Elementen. Beim Umgang mit solchen Arrays ist es wichtig zu wissen, aus wievielen Elementen ein solches Array maximal bestehen soll, damit der Computer ausreichend Platz für die Speicherung der Elemente anlegen kann. Hierfür dient der Dimensionierungsbefehl DIM.

Ein Array A(X,Y) kann allgemein in folgender Tabellenform skizziert werden:

	Spalte 0	Spalte 1	Spalte 2	Spalte 3	...	Spalte Y
Zeile 0	A(0,0)	A(0,1)	A(0,2)	A(0,3)	...	A(0,Y)
Zeile 1	A(1,0)	A(1,1)	A(1,2)	A(1,3)	...	A(1,Y)
Zeile 2	A(2,0)	A(2,1)	A(2,2)	A(2,3)	...	A(2,Y)
Zeile 3	A(3,0)	A(3,1)	A(3,2)	A(3,3)	...	A(3,Y)
:	:	:	:	:		:
Zeile X	A(X,0)	A(X,1)	A(X,2)	A(X,3)	...	A(X,Y)

Beachten Sie, daß die Numerierung der Zeilen und Spalten eines Arrays immer bei Null beginnt. Lautet das höchste Element eines Arrays auf A(3,3), so besteht das Array aus 16 Elementen. Jedes Element trägt dabei einen individuellen Array-Namen. Somit muß bei diesem Array genügend Speicherplatz für 16 mögliche Elemente geschaffen werden.

Arrays können ebenso benannt werden wie einfache Variablen. Sie müssen allerdings mit einem Index versehen sein, der selbst aus einer Variablen oder einem numerischen Ausdruck bestehen darf. Dieses erlaubt, programmtechnisch alle Elemente eines Arrays innerhalb einer Programmschleife ansprechen zu können, wobei dem als Variable ausgelegten Index mit jedem Schleifendurchlauf ein neuer Wert zugewiesen wird, der eben das nächste Element des Arrays selektiert.

A(X,Y), KA(J), B1(J,K) sind solche Arrays mit allgemeinen und damit variablen Indizes. A(1,3), KA(6), B1(5,7) wären spezielle Elemente dieser allgemeinen Arrays.

Die maximale Elementanzahl eines Arrays kann 65535 betragen, da die Indizes im Wertebereich 0 .... 255 liegen dürfen und maximal zwei Dimensionen je Array möglich sind (255x255=65535).

Der von den Elementen eines Arrays belegte Speicherplatz kann mit dem Befehl ERASE wieder freigegeben werden. Die Standard-Variablen A .. Z bzw. A$ .. Z$ werden dadurch nicht gelöscht.

Die maximale Größe eines Arrays beträgt 64 KByte.

d) Standard-String-Variablen

Für die standardmäßigen String-Variablen gelten die bereits bei den numerischen Standard-Variablen genannten Regeln. Sie müssen jedoch mit einem nachgestellten Dollarzeichen versehen sein.

Beispiel: A$, C$, P$, @$(2)

Standard-String-Variable können maximal 16 Zeichen aufnehmen.

e) Einfache String-Variablen

Einfache String-Varibalen müssen denselben Regeln genügen wie einfache numerische Variablen. Sie werden von diesen durch ein nachgestelltes Dollarzeichen unterschieden.

Beispiele: AB$, C7$, Z9$, XX$

Einfache String-Variable können maximal 16 Zeichen aufnehmen.

f) String-Arrays

Die Regeln für die Benennung von String-Arrays entsprechen denen für numerische Arrays. Zur Unterscheidung werden ihren Namen jedoch ein Dollarzeichen nachgestellt, das aber noch vor dem Index auftauchen muß.

Beispiele: A1$(3), KS$(5,5), C9$(J), JH$(P,Q)

Wird nichts weiteres vereinbart, können String-Arrays nur bis zu 16 Zeichen umfassende Strings aufnehmen. In Verbindung mit dem DIM-Befehl läßt sich diese Anzahl variieren. Von dieser Möglichkeit sollte bei der Deklaration von String-Arrays reger Gebrauch gemacht werden, da sich hierdurch viel Speicherplatz einsparen läßt, der sonst durch ungenutzte Reservierung für Strings verloren ginge. Ein Rechenbeispiel möge dieses verdeutlichen.

Angenommen, es sei ein Array mit 25 Elementen deklariert worden, z.B. AB$(4,4). Ohne Spezifizierung der Länge der aufzunehmenden Strings wird nun Speicherplatz für 25 Strings zu je 16 Zeichen reserviert. Kommen im Programm aber nur Belegungen des Arrays mit Strings von 3 Zeichen Länge vor, so wird unnötigerweise ein Speicherplatz von (16-3)*25 = 325 Bytes vergeudet.

10.4 Ausdrücke und Operatoren

Kombinationen von Variablen und Konstanten mittels Operatoren nennt man Ausdrücke (Terme). Je nach Art dieser Größen sind numerische Ausdrücke und String-Ausdrücke zu unterscheiden. Im einfachsten Falle kann ein Ausdruck nur aus einer Variablen oder einer Konstanten bestehen.

Operatoren werden durch spezielle Symbole repräsentiert, die die gewünschte Verknüpfungsart zwischen den Variablen und Konstanten eines Ausdruckes kennzeichnen. Im BASIC unterscheidet man dabei je nach Art der Operationen folgende vier Kategorien:

a) arithmetische Operatoren
b) Vergleichsoperatoren
c) logische Operatoren
d) funktionale Operatoren

Diese Kategorien seien nun nachfolgend näher beschrieben:

a) Arithmetische Operatoren

Arithmetische Operatoren beschreiben die Verknüpfungsarten von numerischen Größen, also die jeweiligen Rechenarten. Treten in einem Ausdruck mehrere Operatoren auf, muß Klarheit darüber herrschen, welche Operation zuerst durchzuführen ist, um das richtige Ergebnis zu erhalten. (Denken Sie beispielsweise an den Grundsatz: Punktrechnung geht vor Strichrechnung). Dazu wird den Operatoren eine bestimmte Vorrangstellung zugewiesen, die in folgender Prioritäten-Liste zusammengestellt werden kann:

Priorität	Operator	Durchzuführende Operation	Beispiel
höchste	^	Potenzbildung	X^3
	-	Negation	-X
	* /	Multiplikation, Division	X*2, X/Y
	MOD	Restbildung	A MOD B
	\	Ganzzahldivision	A\B
niedrigste	+ -	Addition, Subtraktion	X+A, Y-B

Die Reihenfolge dieser Prioritäten kann mit Hilfe von Klammern durchbrochen werden, die immer die höchste Priorität besitzen.

Nachstehend seien als Beispiel einige algebraische Ausdrücke und deren äquivalente Notation in der BASIC-Schreibweise gezeigt:

Normale (mathematische) Notation	BASIC-Ausdruck
x^2	X^2
$(x+3)^4$	(X+3)^4
$2(x+y)$	2*(X+Y)
b^{x^2}	B^X^2

Folgen gleichrangige Operatoren im selben Ausdruck mehrfach hintereinander, werden sie in der Reihenfolge ihres Auftretens von links nach rechts ausgewertet.

3^2^4 bedeutet $(3^2)^4$

Tauchen Potenzen mit negativem Vorzeichen auf, so erfolgt die Auswertung von rechts nach links:

-4^2 bedeutet $-(4^2)$
3^-2 bedeutet 3^{-2}

b) Vergleichsoperatoren

Vergleichsoperatoren erlauben, wie ihr Name es bereits andeutet, die Werte zweier Variablen oder Konstanten miteinander zu vergleichen und damit zu überprüfen, ob eine bestimmte Bedingung vorliegt. Das Ergebnis eines solchen Vergleiches kennt nur die zwei Zustände "wahr" (1) oder "falsch" (0). Dieses zweiwertige (sprich binäre) Resultat, das man auch als boolesches Resultat bezeichnet, erlaubt einem Programm eine Entscheidung darüber zu treffen, mit welchem Unterprogramm bzw. welchem Programmteil die Programmausführung nach dem Vergleich fortgesetzt werden soll. Die Anweisungen IF..GOTO und IF..THEN..ELSE sind solche, die in die Gruppe der bedingten Anweisungen gehören und einen variablen Programmablauf ermöglichen.

Folgende Vergleichsoperatoren können benutzt werden:

Operator	Vergleich auf	Beispiel
=	Gleichheit	A=B
<>	Ungleichheit	X<>Y
<	kleiner als	A<B
>	größer als	B>A
<=	kleiner als oder gleich	Y<=3
>=	größer als oder gleich	X>=1

Anmerkungen:

- Das Gleichheitszeichen wird sowohl bei Vergleichen als auch bei Zuweisungen (siehe LET-Befehl) verwendet. Damit ist es im einen Fall ein Vergleichsoperator und im anderen ein Zuweisungsoperator. Beides ist aber etwas völlig anderes.

- Vergleiche können zwischen zwei numerischen Ausdrücken oder Variablen bzw. String-Ausdrücken oder String-Variablen stattfinden. Ein Vergleich von numerischen Ausdrücken mit String-Ausdrücken ist nicht möglich.

- Ein Vergleich zweier Strings erfolgt Zeichen für Zeichen von links nach rechts gehend unter Betrachtung der Zeichen-Codes (Siehe Anhang C). Sofern beide Strings dieselben Zeichen in gleicher Reihenfolge enthalten und von derselben Länge sind, sind die beiden Strings einander gleich. Strings gleicher Länge, die sich an einer oder mehreren Positionen in den dortigen Zeichen voneinander unterscheiden, sind ungleich. Es gilt dann der String als größer, dessen Zeichen des ersten abweichenden Zeichenpaares den höchsten ASCII-Code besitzt. Wird ein kurzer String mit einem längeren String verglichen und ist der kürzere String zu Beginn des längeren Strings vollständig enthalten, so gilt der kurze String als kleiner. Eine Leerstelle in einem String wird als Zeichen mit dem ASCII-Code &20 (32 dezimal) geführt. Der kürzeste mögliche String ist der Leer- bzw. Nullstring, dessen Vergleichswert dem Zeichen mit dem Code 0 entspricht.

- Arithmetische Operatoren besitzen eine höhere Priorität als die Vergleichsoperatoren, d.h., bevor ein Vergleich zweier Ausdrücke stattfindet, werden diese zuerst berechnet. Im Falle des Vergleiches (2+3) < (2*3) sind die Klammern also nicht erforderlich. In ähnlichen Fällen sollte man aber der Übersicht halber die Klammerung bevorzugen.

c) Logische Operatoren

Als logische Operatoren können die booleschen Funktionen AND, OR und NOT verwendet werden. Die logischen Operatoren liegen in ihrer Priorität unter den arithmetischen und den vergleichenden Operatoren. Die logischen Operatoren lassen sich dazu verwenden, um aufgrund mehrerer Bedingungen den Ablauf des Programmes entsprechend zu steuern. Hierzu können die Anweisungen IF...GOSUB und IF..THEN..ELSE dienen.

Beispiele: IF A<=32 AND B>=90 THEN 150

In dem Beispiel wird das Programm dann mit der Zeile 150 fortgeführt, wenn beide Bedingungen erfüllt sind, also gleichzeitig A kleiner oder gleich 32 UND B größer oder gleich 90 ist.

IF X<>13 OR Y=0 THEN 50

Diese Anweisung erzwingt die Fortführung des Programmes mit der Zeile 50, sofern entweder X ungleich 13 ist ODER Y den Wert Null hat.

In logischen Operationen, die zwei Integer-Zahlen im Bereich von -32768 bis +32767 einschließen, werden diese Zahlen intern in eine 16 Bit umfassende Zweierkomplementdarstellung umgewandelt und dann der logische Vergleich Bit für Bit gemäß der logischen Operation durchgeführt.

Das Ergebnis der möglichen drei logischen Operationen bezogen auf die Verknüpfung zweier Bits, lautet wie folgt:

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X	NOT X
0	1
1	0

Nachdem die Integer-Werte Bit für Bit gemäß diesen Tabellen verknüpft worden sind, wird dieses binäre 16-Bit-Ergebnis in eine dezimale Entsprechung zurückgewandelt.

Beispiele:

41 AND 27

41 = 000000000101001
27 = 000000000011011 } AND

9 = 000000000001001

41 OR 27

41 = 000000000101001
27 = 000000000011011 } OR

59 = 000000000111011

NOT 3

3 = 0000000000000011
NOT

-4 = 1111111111111100

NOT X kann grundsätzlich auch durch den Ausdruck -(X+1) gebildet werden.

d) Funktionale Operatoren

Funktionen sind Operationen, die aus einem Operanden mit Hilfe geeigneter BASIC-Routinen einen speziellen Wert berechnen. Diese Funktionen (hier nicht auf die mathematische Definition beschränkt gemeint) sind in Kapitel 14 ausführlich beschrieben. Das PC-1600-BASIC stellt folgende Funktionen zur Verfügung:

ABS	Bildung des Absolutbetrages
ACS	Arcus-Cosinus-Funktion
ASN	Arcus-Sinus-Funktion
ATN	Arcus-Tangens-Funktion
COS	Cosinus-Funktion
DEG	Winkelumwandlung von Sexagesimal- in Dezimal-Form
DMS	Winkelumwandlung von Dezimal- in Sexagesimal-Form
EXP	Exponentialfunktion (Potenzierung zur Basis e)
INT	Extraktion des ganzzahligen Anteiles
LN	natürlicher Logarithmus
LOG	dekadischer Logarithmus
PI	Wert der transzendenten Kreiszahl π
RND	Erzeugung einer zufälligen Zahl
SGN	Extraktion des Vorzeichens
SIN	Sinus-Funktion
SQR	Bildung der Quadratwurzel
TAN	Tangens-Funktion

Winkelargumente können in den drei Winkelmaßen angegeben bzw. ausgegeben werden. Welches Winkelmaß gilt, hängt von der Einstellung des Winkelmodus ab:

Winkelmaß	Winkelmodus
Altgrad	DEGREE
Bogenmaß	RADIAN
Neugrad	GRAD

Zusätzlich zu den obigen Funktionen kennt das PC-1600-BASIC noch drei String-Funktionen, mit denen die Manipulation von Strings möglich ist. Eine genaue Beschreibung dieser Befehle LEFT$, MID$ und RIGHT$ ist in Kapitel 14 zu finden.

11. Dateien

Eine Datei besteht aus einer Ansammlung von Daten-Sätzen, die auch Records genannt werden und sich mit Hilfe externer Geräte (z.B. einem Diskettenlaufwerk) auf verschiedenen Medien (z.B. einer Diskette) speichern lassen. Sie können bei Bedarf in den Arbeitsspeicher des Computers gelesen oder aber in die Datei zurückgeschrieben werden, wobei sie stets unter einem gemeinsamen Identifikationsmerkmal, dem Dateinamen, ansprechbar sind. Dateien können Strings oder numerische Daten enthalten und im ASCII-Format oder in binärer Form vorliegen. Sie lassen sich auf Cassetten, Disketten und RAM-Disks speichern und über Interfaces an periphere Geräte bzw. andere Computer übertragen oder von diesen holen.

Alle vom PC-1600 handhabbaren Dateien sind sequentieller Natur. Das bedeutet, daß die einzelnen Daten-Sätze nur nacheinander geschrieben oder geladen werden können. Ein wahlfreier Zugriff auf jeden beliebigen Record ist nicht möglich.

11.1 Datei-Bezeichner

Dateien sind durch sogenannte Datei-Bezeichner identifizierbar. Ein solcher Datei-Bezeichner besteht aus maximal drei Anteilen und besitzt folgendes Format:

d:filename.ext

Diese im englischen Sprachraum gebräuchliche Darstellung läßt sich leider nicht so elegant in ein deutschsprachiges Äquivalent übertragen. Deshalb seien nun die in diesem Handbuch verwendeten "Eindeutschungen" den englischen Begriffen gegenübergestellt:

Kürzel	englischer Begriff	sinngemäße "Eindeutschung"
d	device, device name	logischer Gerätename, Medium: Daten-Quelle/-Ziel
filename	file name, name of file	Datei-Name
ext	extension	Extension, Gruppenseparator

In deutscher Formulierung werden wir dieses Format, je nachdem ob eine Datei gelesen oder geschrieben wird, zwischen folgenden beiden Formaten unterscheiden:

```
Dateibezeichner =  Datenquelle:Dateiname.Extension
Dateibezeichner =  Datenziel:Dateiname.Extension
```

A) Logischer Gerätename (Medium)

Der logische Gerätename spezifiziert, auf welches physikalische Gerät die Datei geschrieben oder von diesem geholt werden soll. Damit bezeichnet der logische Gerätename auch auf indirekter Weise das eigentliche Speichermedium.

Folgende logische Namen sind zulässig:

logischer Name	Gerät
S1:	Modulfach S1
S2:	Modulfach S2
COM1:	RS-232C-Schnittstelle
COM2:	Optoelektronisches Interface (SIO)
CAS:	Cassetten-Recorder
X:, Y:	2,5"-Disketten-Laufwerk

B) Dateiname

Jeder Datei muß ein Name zugewiesen werden, bevor sie gesichert werden kann (eine Ausnahme besteht bei Aufzeichnungen mit einem Cassetten-Recorder, die zum PC-1500 kompatibel sein sollen). Der Name kann dabei aus maximal 8 Zeichen zusammengesetzt sein, die aus folgendem Zeichenvorrat stammen müssen:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
# $ % & ' ( ) - ^ { } @
```

Einer Datei kann immer nur ein Name zugewiesen werden.

C) Extension

Die Datei-Extension dient einem zusätzlichen Idendifikations-merkmal. Sie beschreibt die Zugehörigkeit der Datei zu einer bestimmten Dateien-Gruppe, also beispielsweise ob es sich um ein BASIC-Programm handelt, ein Maschinensprache-Programm usw. Deshalb sprechen wir bei der Extension manchmal auch von einem sogenannten Gruppenseparator. Finanzmathematischen Programmen könnte man beispielsweise die Extension .FIN zuordnen und den Maschinensprache-Programmen die Extension .BIN.

Wird ein BASIC-Programm mit dem Befehl SAVE gesichert, weist der PC-1600 dieser Datei automatisch die Extension .BAS zu. Um ein solches Programm mit dem LOAD-Befehl wieder zu laden, braucht dieser Gruppenseparator nicht angegeben zu werden, da das BASIC standardmäßig diese Extension voraussetzt, sofern kein solcher Zusatz dem Dateinamen beigefügt wird. Werden die Befehle FILES bzw. LFILES benutzt, um ein Inhaltsverzeichnis über die auf dem Medium befindlichen Dateien zu erhalten, erscheinen die BASIC-Programme immer mit der Kennzeichnung .BAS, sofern man bei ihrer Aufzeichnung keine andere Extension spezifiziert hat.

Werden BASIC-Programme kopiert, muß die gültige Extension .BAS (oder die vom Anwender spezifizierte) immer im Dateibezeichner angegeben werden.

11.2 Speichern und Laden von Dateien

Dateien lassen sich auf Cassetten, Disketten oder als RAM-Disks initialisierten Modulen speichern. Je nach Art des verwendeten Speichermediums unterscheiden wir entsprechende Datei-Arten:

A) Cassetten-Dateien

Auf Cassetten lassen sich Dateien nur in sequentieller Anordnung aufzeichnen. Will man beispielsweise auf die fünfte Datei einer Cassette zugreifen, müssen die vorangehenden vier ersteinmal am Tonkopf vorbeigeführt werden, bis anhand der Dateikennung, die aus einem Block an Identifikationsdaten besteht, die fünfte Datei gefunden und mit dem CLOAD-Befehl geladen werden kann. Der Zugriff auf Cassetten-Dateien dauert deshalb im Vergleich zu Disketten-Dateien sehr lange.

Folgende Befehle zur Bedienung des Recorders gibt es:

CHAIN, CLOAD, CLOAD?, CLOADM, CSAVE, CSAVEM,
INPUT#, MERGE, PRINT# und RMT ON/OFF.

Die Anzahl der auf Cassette ablegbaren Dateien läßt sich nicht angeben, da sie von vielen Faktoren abhängt. Dieses sind unter anderen die Bandlänge, der Umfang der Dateien und deren Typ.

B) Disketten-Dateien

Disketten-Dateien werden auf der Diskette (oder dem so wirkenden RAM-Modul) in Form kleiner Sektoren aufgezeichnet. Der Computer ist dabei in der Lage, auf jeden dieser Sektoren individuell und sofort zuzugreifen. Das bedeutet, daß sich eine gewünschte Datei quasi ohne Wartezeit laden läßt.

a) Inhaltsverzeichnisse

Um zu wissen, welche Dateien sich auf einer Diskette befinden, führt der Computer Buch und legt sich auf jeder Diskette ein Inhaltsverzeichnis an. In dieses Verzeichnis trägt er nicht nur die Dateinamen ein, sondern auch noch weitere Informationen, wie z.B. Uhrzeit und Datum der Aufzeichnung. Darüberhinaus vermerkt er auch noch, auf welchen physikalischen Sektoren jede Datei zu finden ist.

Mit Hilfe des FILES-Befehles kann ein solches Inhaltsverzeichnis angezeigt oder über LFILES auf einem Drucker ausgegeben werden. Dabei bleiben die nur für den internen Gebrauch erforderlichen Informationen (wie Lage der Sektoren) geheim. Es werden also nur die Dateinamen mit Datum und Uhrzeit sowie einem eventuellen Schreibschutzattribut ausgewiesen. Das Format lautet hierbei:

```
filename.ext,P 12/31 08:35
               ¦    ¦     ¦
               ¦    ¦     ¦
               ¦    ¦     Uhrzeit (Minute:Sekunde)
               ¦    ¦
               ¦    Datum (Monat:Tag)
               ¦
               Schreibschutz gesetzt mit SET
```

Nachstehende Befehle dienen der Handhabung von Diskettendateien:

BLOAD, BSAVE, CLOSE, COPY, FILES, INIT,
INPUT#, KILL, LFILES, LOAD, MAXFILES, NAME,
OPEN, PRINT#, SAVE und SET.

b) Anzahl der Disketten-Dateien

Die Anzahl der Dateien, die auf einer Diskette oder RAM-Diskette gespeichert werden können, bestimmt der zur Verfügung stehende Speicherplatz für das Inhaltsverzeichnis. Eine Diskette kann bis 48 Dateien pro Seite aufnehmen, ein RAM-Modul des Typs CE-1600M maximal 48 Dateien.

HINWEIS: CE-159-Module können nur benutzt werden, um einfache Programme darauf zu speichern, die sich nicht in der Form wie Disketten-Dateien ansprechen lassen. Damit können diese Programme nicht mit dem LOAD-Befehl geladen werden. Nachdem ein solches Modul in eines der beiden Modulfächer eingesetzt worden ist, kann es mit dem Befehl TITLE aktiviert und über sein Programm mit RUN gestartet werden, so als ob es in den internen Arbeitsspeicher geladen sei.

c) Mehrdeutige Datei-Bezeichner

Will man mit einem Befehl nicht eine individuelle Datei, sondern eine Gruppe von Dateien ansprechen, so läßt sich dieses durch die Anwendung von sogenannten Jokern (**wildcards**) erreichen, die man in einem gemeinsamen Datei-Bezeichner stellvertretend für jedes erdenkliche, aber zulässige Zeichen einsetzt.

Eines dieser mehrdeutigen Zeichen ist das Fragezeichen ?, das an jede beliebige Position des Dateinamens oder der Extension eingesetzt werden kann und jeweils das an dieser Position gedachte Zeichen ersetzt. Der andere und weitaus mächtigere Joker ist der Multiplikations-Stern. Er vertritt nicht nur ein einzelnes Zeichen, sondern entweder einen kompletten Dateinamen oder eine Extension.

Die Wirkung des Joker-Einsatzes möge die nachstehende Tabelle veranschaulichen:

Mehrdeutiger Bezeichner	Mögliche passende Dateien
?ET	SET, MET, RET, @ET, 4ET
?PT?	RPT1, RPT2, SPTM, (PT), -PTS
??????	beliebiger Name mit sechs Zeichen
*	beliebiger Name mit acht Zeichen
SHARP.*	jede Datei mit dem Namen SHARP
.	jede beliebige Datei

Die Spezifikation mehrdeutiger Datei-Bezeichner resultiert bei Verwendung der Befehle FILES und LFILES darin, daß alle Datei-Bezeichner, die in diese Maske passen und einer existierenden Datei entsprechen, aufgelistet werden. Der Bezeichner *.* sorgt dafür, daß alle existierenden Dateien bei dieser Auflistung erwähnt werden.

11.3 Schreibschutz

Jede auf Diskette oder RAM-Disk befindliche Datei kann gegen unbeabsichtigtes Beschreiben mit einem Schreibschutz versehen werden. (Siehe hierzu die Beschreibungen des SET-Befehles.)

Außer dieses softwaremäßigen Schutzes lassen sich Dateien auch hardwaremäßig vor unzulässigem Beschreiben bewahren. Hierfür sind sowohl die Disketten als auch die Module mit sogenannten Schreibschutzschaltern ausgestattet. Bei RAM-Modulen ist dieser Schalter bei ausgeschaltetem Computer in Stellung ON zu bringen, um den Schreibschutz zu aktivieren. Disketten weisen für jede Seite einen Schalter auf, der mit einen spitzen Gegenstand so zu verschieben ist, daß er aus dem Sichtfenster verschwindet.

Darüberhinaus läßt sich der Speicherinhalt des Computers gegen den unerlaubten Zugriff unberechtigter Personen durch die Vergabe eines Kennwortes (pass word) schützen. Der Inhalt des damit geschützten Speichers kann dann weder gelöscht, verändert noch aufgelistet werden, bevor nicht durch die Eingabe des richtigen Kennwortes der Schutz aufgehoben wird.

11.4 Erstellung einer Datei

Um eine Datei zu erzeugen, sollte nach den folgenden Schritten vorgegangen werden:

a) Geben Sie den Befehl MAXFILES ein, um die Anzahl der gleichzeitig offen sein dürfenden Dateien festzulegen.

b) Geben Sie den Befehl OPEN ein, um die Zuordnung zwischen einem Datei-Namen und einer Datei-Nummer herzustellen. Öffnen Sie dann die Datei mit dem Befehl OUTPUT.

c) Schreiben Sie nun die gewünschten Daten mit dem Befehl PRINT# in die Datei.

d) Schließen Sie dann die Datei mit dem Befehl CLOSE. Dies ist erforderlich bevor die Datei erneut geöffnet werden kann, um beispielsweise die Daten mit dem Befehl INPUT# zu lesen.

Beispiel:

Das nachfolgende kleine Programm erzeugt auf der Diskette eine Datei, in der die Namen und Adressen von Personen gespeichert werden. Es handelt sich also um ein kleines elektronisches Adreßbuch.

```
10 MAXFILES=1
20 OPEN "X:ADRESSE" FOR OUTPUT AS #1
30 INPUT "NAME    = ";N$
40 IF N$="ENDE" THEN 100
50 INPUT "STADT   = ";S$
60 INPUT "TELEFON = ";T$
70 PRINT #1,N$;",";S$;",";T$
80 PRINT
90 GOTO 30
100 CLOSE #1
110 END
```

```
Zeile  10  erlaubt die Öffnung einer einzelnen Datei.
Zeile  20  öffnet diese Datei zum Zwecke des Schreibens.
Zeile  30  fragt nach dem Personen-Namen.
Zeile  40  springt zu Zeile 100, falls anstelle des
           Namens das Wort ENDE eingegeben wurde.
Zeile  50  Fragt nach der Stadt, also dem Wohnort.
Zeile  60  fragt nach der Telefon-Nummer.
Zeile  70  schreibt einen Datensatz in die Datei.
Zeile  80  bildet auf dem Display einen Abstand zu den
           bisher eingegebenen Daten.
Zeile  90  veranlaßt die Abfrage weiterer Daten.
Zeile 100  schließt die Datei.
Zeile 110  beendet das Programm.
```

Geben Sie dieses Programm in Ihren PC-1600 ein und starten Sie es mit RUN. Dann passiert folgendes:

In der Anzeige erscheint die Aufforderung: NAME = ?

Geben Sie nun den Namen irgendeiner Person ein, deren Wohnort und deren Telefonnummer Sie zu speichern gedenken. Schließen Sie diese Eingabe mit Betätigung der Taste ENTER ab.

In der Anzeige erscheint nun die Meldung : STADT = ?

Tippen Sie hierauf den Namen der Stadt, also den Wohnort ein und schließen Sie auch diese Eingabe mit ENTER ab.

Die Anzeige stellt nun folgende Frage : TELEFON = ?

Geben Sie jetzt die entsprechende Telefonnummer ein. Schließen Sie auch diese Eingabe (wie üblich) mit der Taste ENTER ab.

Draufhin beginnt ein neuer Fragezyklus und Sie können die Daten einer weiteren Person eingeben und speichern lassen. Möchten Sie jedoch das Programm beenden, so geben Sie anstelle des Namens einfach das Wort ENDE ein. Der Programmablauf wird dadurch in Zeile 100 verzweigt und dadurch die Datei geschlossen und das Programm mit Zeile 110 beendet.

Falls Ihnen bei der Probe mit diesem Programm auf die Schnelle keine geeigneten Namen usw. einfallen, nehmen Sie einfach die Daten nachfolgender Liste:

M.JOHNES	NEW YORK	212-758-0354
T.SMITH	LONDON	01-634-4431
M.BERRY	NEW YORK	212-432-0012
N.ITO	TOKYO	03-927-1345
S.SHARP	HAMBURG	040-677-321
P.PETERS	LONDON	01-433-0056

Lassen Sie sich anschließend ein Inhaltsverzeichnis aller auf Diskette befindlichen Dateien anfertigen (FILES *.*), sollten Sie in diesem die eben erstellte Datei ADRESSE.BAS wiederfinden.

HINWEIS: Wenn Sie das Programm erneut starten, wird die Datei in Zeile 20 neu eröffnet. Dieses hat zur Folge, daß sämtliche zuvor in dieser Datei aufgenommenen Daten verloren gehen.

Der einzige Weg, neue Daten in eine existierende Datei schreiben zu können, besteht durch ihre Öffnung im APPEND-Modus. Hierfür muß ein separates Programm erstellt werden, welches folgende Anweisung enthält:

```
OPEN "X:ADRESSE" FOR APPEND AS #1
```

Der APPEND-Befehl kann nur im Zusammenhang mit Disketten-Dateien verwendet werden, also nur bei solchen Dateien, die sich auf einer Diskette oder einem als RAM-Disk initialisierten RAM-Modul befinden.

Bei sequentiellen Dateien können zusätzliche Daten immer nur an das Dateiende angefügt, niemals aber in die bereits bestehenden Datei-Sätze eingefügt werden. Eine solche Einfügung kann jedoch mit einem geeigneten Programm auf die Art erreicht werden, daß man alle Daten der Datei bis zur gewünschten Position einliest, den neuen Daten-Satz anfügt, die restlichen Daten-Sätze einliest und alles in eine neue Datei zurückschreibt.

11.5 Zugriff auf Dateien

Um auf eine Datei zuzugreifen sind folgende drei Schritte in der genannten Reihenfolge einzuhalten:

a) Bestimmen Sie mit dem Befehl MAXFILES die Anzahl der zu öffnenden Dateien.

b) Öffnen Sie die Datei mit dem OPEN-Befehl im INPUT-Modus, um Datensätze von der Datei lesen zu können.

c) Lesen Sie mit dem INPUT#-Befehl die Daten aus der Datei satzweise in die erforderlichen Variablen ein.

d) Schließen Sie die Datei mit dem CLOSE-Befehl. Dieses ist erforderlich, um mit einer Neueröffnung im APPEND-Modus neue Datensätze an die Datei anfügen zu können.

Jedesmal wenn man eine Datei zum Lesen öffnet, wird ein interner Zeiger auf den Dateianfang gesetzt. Mit dem ersten folgenden INPUT#-Befehl wird dann der erste Datensatz eingelesen und der interne Zeiger auf den nächsten Datensatz weitergesetzt. Mit dem kommenden INPUT#-Befehl wird dann dieser Datensatz gelesen und so fort, bis das Ende der Datei erreicht ist.

Wird in dem Programm, das die Datei liest, keine Überprüfung eingebaut, ob das Dateiende erreicht ist, und über das Dateiende deswegen hinausgelesen, bricht das Programm mit Ausgabe des ERROR-Codes 165 ab.

Beispiel:

Das folgende Programm liest die Daten der Datei ADRESSE, die mit dem vorangegangenen Programm erstellt worden ist, und listet die Datensätze auf, die den Wohnort "NEW YORK" beinhalten:

```
10 MAXFILES=1
20 OPEN "X:ADRESSE" FOR INPUT AS #1
30 PRINT "NEW YORK":PRINT
40 PRINT "NAME","TELEFON-NR."
50 IF EOF(1) THEN 100
60 INPUT #1,N$,S$,T$
70 IF S$="NEW YORK" THEN 90
80 GOTO 50
90 PRINT N$,T$ : GOTO 50
100 CLOSE #1
110 END
```

```
Zeile  10  erlaubt die Öffnung einer einzelnen Datei.
Zeile  20  öffnet die Datei zum Lesen der Records.
Zeile  30  gibt einen Listenkopf aus.
Zeile  40  gibt eine Listenüberschrift aus.
Zeile  50  prüft, ob das Dateiende erreicht ist. Trifft
           dieses zu, erfolgt ein Sprung zu Zeile 100.
Zeile  60  liest einen Datensatz ein.
Zeile  70  überprüft, ob der Wohnort NEW YORK lautet.
           Wenn ja, wird mit Zeile 90 fortgefahren.
           (Die Schreibweise muß mit der der Aufzeichnun
           exakt übereinstimmen !)
Zeile  80  sorgt für das Lesen des nächsten Datensatzes.
Zeile  90  zeigt die gelesenen Daten an.
Zeile 100  schließt die Datei.
Zeile 110  beendet das Programm.
```

Geben Sie dieses Programm in den PC-1600 ein und starten Sie es mit RUN. In der Anzeige erscheint dann:

```
NEW YORK

NAME          TELEFON-NR.
M.JONES       212-758-0354
M.BERRY       212-432-0012
```

Das obige Programm kann durch einfache Änderungen so modifiziert werden, daß ein variables Suchkriterium möglich ist:

```
25 INPUT"STADT = ";X$
30 PRINT X$
70 IF S$=X$ THEN 90
```

11.6 Änderung einer Datei

Wie zuvor gesagt, lassen sich bei einer sequentiellen Datei neue Daten nur an das Dateiende anfügen, wenn diese zuvor im APPEND-Modus geöffnet worden ist. Um Datensätze innerhalb einer Datei einfügen zu können, bedarf es einer anspruchsvolleren Programmtechnik, als wir es in den Programm-Beispielen zeigen konnten.

Eine Änderung der Zeile 20 des ersten Beispielprogrammes, ermöglicht den Anhang neuer Daten an unsere Datei ADRESSE:

```
20 OPEN "X:ADRESSE" FOR APPEND AS #1
```

Die Funktionen DSKF, EOF, LOC, LOF und MAXFILES stehen für den Aufbau von komplexen Dateihandhabungs-Programmen zur Verfügung, um Dateigrößen zu überprüfen, sicherzustellen, daß die Diskette noch nicht voll ist oder andere Fehler abgefangen werden usw.

12. Zugriff auf die seriellen Ports

Die beiden eingebauten seriellen Schnittstellen, auch Interfaces oder Ports genannt, gestatten es, den PC-1600 an andere Computer oder verschiedene Peripheriegeräte anzuschließen und mit diesen zu kommunizieren, d.h. Daten auszutauschen. Einer dieser Ports entspricht dem weitverbreiteten RS-232C-Standard. Der andere ist auf zukünftige Entwicklungen ausgerichtet und erlaubt den Datenaustausch über Glasfaserkabel.

12.1 Auswahl der Schnittstelle

Beide Interfaces lassen sich über geeignete BASIC-Befehle durch logische Benennungen, die als Parameter den Befehlen beizufügen sind, ansprechen. Diese Benennungen lauten:

COM1:	RS-232C-Interface (Standard-Interface)
COM2:	SIO-Interface (optoelektronisches Interface)
COM:	über SETDEV selektiertes Interface

Wird einer der beiden seriellen Ports mittels SETDEV und einem der zugehörigen Paramter COM1: oder COM2: selektiert, ist dieser für eine Datenkommunikation geöffnet. Es können nun Daten über dieses Interface empfangen oder gesendet werden. Die Selektion gilt solange, bis mit einem neuen SETDEV-Befehl etwas anderes vereinbart wird.

Folgende BASIC-Befehle stehen zur Bedienung der seriellen Ports und damit für die Daten-Kommunikation bereit:

COMn ON/OFF/STOP, INIT, PCONSOLE, ON COMn GOSUB,
ON PHONE GOSUB, OUTSTAT, INSTAT, PZONE, RCVSTAT,
PHONE ON/OFF/STOP, SETCOM, SETDEV, SNDBRK, SNDSTAT,

Zusätzlich lassen sich Daten auch mittels einiger Druckbefehle über die seriellen Schnittstellen aussenden. Nach Einschaltung des Computers werden die Ausgaben der Befehle LLIST, LFILES und LPRINT standardmäßig an den Drucker weitergeleitet. Mit den drei Optionen des SETDEV-Befehles besteht jedoch die Möglichkeit, die Ausgaben auf eine der Schnittstellen umzuleiten oder wieder auf den Drucker zurückzuführen. Bei einer Rücksetzung der Ausgaben auf den Druckerkanal werden die Ports automatisch geschlossen. Um komplette Dateien über die Schnittstellen laden oder sichern zu können, sind die Befehle LOAD und SAVE bereitgestellt.

12.2 Protokoll-Optionen

Beim Datentransfer kommt es häufig vor, daß der Empfänger gerade keine weiteren Daten annehmen kann, weil er noch mit der Ablage oder der Verarbeitung der bisher gelieferten Daten oder anderen Dingen beschäftigt ist. In einer solchen Situation gingen alle weiterhin gesendeten Daten verloren. Um dieses zu vermeiden, müssen sich Sender und Empfänger darüber verständigen können, wann der Datentransfer zu stoppen und wieder fortzusetzen ist. Einen solchen Informationsaustausch, der die eigentliche Datenübertragung steuert, nennt man "Handshaking". Dieses Handshaking kann entweder über Sonder-Zeichen per Software oder aber über spezielle Steuersignale per Hardware abgewickelt werden. Damit sich die kommunizierenden Geräte auch wirklich verstehen, müssen sie sich an dasselbe Handshake-Verfahren, das sogenannte Protokoll halten. Der PC-1600 ist darauf ausgelegt, hardwaremäßige Handshakes durchzuführen.

Bevor ein Sende- oder Empfangsvorgang über einen seriellen Port erfolgen kann, muß zuerst das Übertragungsprotokoll vereinbart werden. Dieses Protokoll beschreibt, wie die Steuersignale des Ports gesetzt sein müssen, um den Datenaustausch überwachen und steuern zu können. Die einzelnen Handshake-Protokolle für das Senden und den Empfang sind über die Befehle SNDSTAT und RCVSTAT einstellbar. Für den Sendevorgang können darüberhinaus mit dem OUTSTAT-Befehl alle Steuersignale individuell gesetzt werden. Das Setzen dieser Signale erfordert die genaue Kenntnis über den Ablauf des Datenaustausches. Werden keine OUTSTAT-Parameter vereinbart, führt der PC-1600 eine einfache Daten-Übertragung aus. Das derzeit eingestellte Empfangsprotokoll kann über den Befehl INSTAT abgefragt werden.

12.3 Kommunikations-Parameter

Ein Datenaustausch zwischen dem Computer und anderen Geräten ist nur dann erfolgreich, wenn sie sich an dieselbe Übertragungsgeschwindigkeit halten und auch in anderen Übertragungs-Parametern übereinstimmen. Die einstellbaren Kommunikations-Parameter sind:

a) Übertragungsgeschwindigkeit : 50 bis 38400 baud
b) Länge eines Datenwortes : 5 bis 8 bits
c) Parität : gerade, ungerade, keine
d) Anzahl der Stopp-Bits : 1 oder 2
e) XON/XOFF-Protokoll : an oder aus
f) SHIFT IN/SHIFT OUT Protokoll: an oder aus

Die Einstellung dieser Parameter läßt sich über SETCOM vornehmen und mit COM$ abfragen.

Bei Anwendung der Befehle SAVE, LOAD, BSAVE und BLOAD oder bei einem Datentransfer zwischen einer RAM-Disk und einem Interface mittels COPY beträgt die maximale Übertragungsgeschwindigkeit 9600 baud beim RS-232C-Port (COM1:) und 38400 baud beim SIO-Interface (COM2:). Bei Verwendung der Befehle INPUT, INPUT#, PRINT#, LLIST und LPRINT beträgt die maximale Übertragungsrate für beide Ports 4800 baud.

12.4 Empfangs-Puffer

Daten, die über eines der beiden seriellen Interfaces empfangen werden, speichert der Computer zuerst in einem Empfangs-Puffer. Die Größe dieses Zwischenspeichers läßt sich mit dem Befehl INIT vereinbaren und damit den individuellen Bedürfnissen anpassen.

12.5 Ausgabe über einen seriellen Port

Für die Ausgabe von Daten über die seriellen Schnittstellen stehen die folgenden BASIC-Befehle bereit:

CHR$, LFILES, LLIST, LPRINT, LPRINT USING, OPEN, PCONSOLE, PRINT#, PRINT# USING, PZONE und SAVE.

A) Senden von Programmen und Daten

Daten, die durch ein Programm oder dessen Auflistung erzeugt werden, lassen sich über die seriellen Schnittstellen nach der folgenden Prozedur ausgeben:

a) Öffnen Sie das gewünschte Interface mittels SETDEV-Befehl mit Angabe der PO-Option, damit die Ausgaben von LPRINT, LLIST und LFILES zu diesem Port umgeleitet werden können.

b) Bestimmten Sie mit dem Befehl PZONE das Ausgabeformat für die LPRINT-Anweisung.

c) Setzen Sie mit PCONSOLE die passende Zeilenlänge und den gewünschten Zeilenend-Code.

Diese Vorbereitungen erlauben es, Programme an andere Computer zu übertragen. Laden Sie das gewünschte Programm zuerst mit dem LOAD-Befehl in den Arbeitsspeicher. Verwenden Sie danach den SETDEV-Befehl, um die Datenausgabe auf die RS-232C-Schnittstelle zu leiten und geben Sie zum Schluß den LLIST-Befehl ein. Das komplette Programm wird dann Zeile für Zeile über den seriellen Port ausgegeben.

Dieser Port kann auch als logische Datei eröffnet werden, so daß mit dem PRINT#-Befehl sequentielle Datensätze über diesen Port ausgegeben werden können. Halten Sie sich dazu bitte an folgende Schritte:

a) Legen Sie mit MAXFILES die Anzahl der maximal zu öffnenden Dateien fest.

b) Öffnen Sie den Port mit Hilfe des Befehles OPEN für eine Ausgabe.

c) Benutzen Sie innerhalb eines Programmes entweder die Anweisung PRINT# oder aber PRINT# USING, um Datensätze auf den Port zu schreiben, so als ob es sich bei diesem um eine Datei handelte.

d) Schließen Sie nach Ausgabe aller Datensätze den Port, so wie Sie es mit einer normalen Datei auch tun würden, mit Hilfe des Befehles CLOSE.

B) Senden von Programm-Dateien

Eine auf Diskette oder RAM-Diskette befindliche Datei läßt sich direkt über einen seriellen Port ausgeben, wenn dazu der SAVE-Befehl in Anspruch genommen und so verwendet wird, als wolle man eine Datei auf Diskette oder RAM-Disk sichern.

C). Senden von Steuer-Codes

Mit Hilfe der CHR$-Funktion lassen sich auch nichtdarstellbare Zeichen, d.h. Steuer-Codes über einen Port ausgeben, so wie man es auch bei der Ausgabe über den Drucker bewerkstelligen kann. So dient der ASCII-Code &04 manchen Computern als Signal für das Ende eines Textes (EOT). Dieser Code kann, wenn der Port zuvor über den Befehl SETDEV für eine Ausgabe vorbereitet worden ist, mit folgender BASIC-Anweisung über die Schnittstelle ausgegeben werden:

```
10 LPRINT CHR$(4)
```

12.6 Empfang über einen seriellen Port

Folgende Kommandos und Funktionen stehen zur Verfügung, um von einem der beiden seriellen Ports Daten zu empfangen:

INIT, INPUT#, LOAD, OPEN, PCONSOLE, RXD$, SNDBRK.

A) Empfang von Daten

Die an einem der seriellen Ports ankommenden Daten können unter Einhaltung folgender Prozedur eingelesen werden:

a) Bestimmen Sie mit MAXFILES die Anzahl der zu öffnenden Dateien.

b) Setzen Sie die erforderliche Größe des Empfangs-Puffers mittels INIT.

c) Öffnen Sie den Port im INPUT-Modus mit dem Befehl OPEN.

d) Lesen Sie die am Port empfangenen Daten satzweise mit dem Befehl INPUT# ein, so als ob es sich bei dem Port um eine ganz normale Datei handelte.

e) Schließen Sie nach Empfang aller Daten den Port mittels CLOSE-Befehl.

B) Empfang von Dateien

Eine Datei kann über ein serielles Interface empfangen und im Arbeitsspeicher des Computers abgelegt werden, wenn man hierfür den LOAD-Befehl benutzt. Dieser ist dabei genau so anzuwenden, als ob man eine Datei von einer Diskette oder RAM-Disk laden wolle.

13. Fehler-Korrektur

Ein Programm zu schreiben, das auf Anhieb perfekt läuft, ist so gut wie unmöglich. Trotz sorgfältigster Arbeitsweise wird man kaum das Auftreten von Tippfehlern vermeiden können. Selbst bei erfahrenen Programmierern findet man hie und da Regelverstöße gegen die Befehlssyntax. Auch wenn man bei der Programmeingabe diese beiden Fehlerarten wirklich auszuschließen vermag, so ist man noch lange nicht gegen Fehler gefeit, die den strukturellen und logischen Aufbau betreffen. Kurz gesagt: Zum Programmieren gehört auch stets das Editieren.

Deshalb sei im folgenden Abschnitt der Prozeß beschrieben, wie man sich sinnvoll und systematisch dieser Fehler entledigt, bis ein funktionsfähiges Programm entsteht.

13.1 Syntax-Fehler

Startet man ein neu erstelltes Programm, wird der Computer die darin enthaltenen Syntax- sowie meisten anderen Fehler nacheinander Zeile für Zeile entdecken. Sobald ein Fehler erkannt ist, unterbricht der Computer das laufende Programm und zeigt einen ERROR-Code (siehe hierzu Anhang F) sowie eine Zeilennummer an, anhand derer sich Art und Ort des Fehlers erkennen lassen.

Liegt ein Syntax-Fehler (ERROR 1) vor, sollten Sie am besten im PRO-Modus den Befehl LIST eingeben und dabei als Parameter die Zeilennummer verwenden, die in der Fehlermeldung genannt ist. Dadurch wird die fehlerbehaftete Zeile auf dem Display angezeigt und kann mit den Editier-Tasten korrigiert werden. Sollte sich bei einem erneuten Programm-Start ein weiterer Syntax-Fehler im Programm entdecken lassen, so verfahren Sie entsprechend. Damit wird das Programm schrittweise von Syntax-Fehlern befreit.

Andere Fehler können zwar auf ähnlichem Wege beseitigt werden, bedürfen aber meist doch einer anderen Vorgehensweise zum Aufspüren der eigentlichen Fehlerursache. So mag zum Beispiel in einer mit FOR....NEXT zu bildenden Programmschleife das Befehlswort NEXT vergessen worden sein. In diesem Falle hilft es also wenig, sich die Zeile, in der sich der Fehler bemerkbar macht, anzeigen zu lassen. Vielmehr muß durch die Betrachtung mehrerer Zeilen herausgefunden werden, an welcher Stelle der NEXT-Befehl vergessen worden ist. Allein die Praxis wird Ihnen also die geeigneten Wege zur Fehlerbeseitigung lehren können.

Syntax- und Schreibfehler sind mit den in Kapitel 9 erläuterten Editier-Funktionen korrigierbar. Diese erlauben, wie Sie selbst feststellen werden, den geringsten Korrektur-Aufwand.

13.2 Trace-Modus

Mit den beiden Kommandos TRON (trace mode on) und TROFF (trace mode off) besteht die Möglichkeit, eine Programmablaufverfolgung ein- bzw. auszuschalten. Bei eingeschaltetem Trace-Modus wird nach Ausführung einer jeden Programmzeile der Programmablauf für die Dauer einer halben Sekunde unterbrochen und im Display auf der rechten Seite die Nummer dieser Zeile angezeigt, bevor mit der Abarbeitung der nächsten Zeile fortgefahren wird. Damit kann verfolgt werden, ob ein Programm tatsächlich zu den gewünschten Unterroutinen oder anderen Programmteilen verzweigt oder andere unvorhergesehene Wege geht. Der Trace-Modus läßt sich auch so einstellen, daß eine schrittweise Abarbeitung von Zeilen per Knopfdruck auf die Taste [↓] möglich ist. Der Trace-Modus bleibt solange aktiv, bis das Kommando TROFF eingegeben wird.

Als weiteres Hilfsmittel zur Lokalisierung von Fehlern dient das STOP-Kommando, das strategisch vor den als fehlerhaft erkannten Programmteil gesetzt werden sollte. An dieser Stelle bricht dann der Programmablauf unter Ausgabe einer entsprechenden Meldung, die mit der Angabe der Zeilennummer versehen ist, ab. Nun lassen sich vielfältige Untersuchungen angestellen, um Rückschlüsse auf die Fehlerursache zu ziehen. Eine Fortführung des unterbrochenen Programmes ist durch Eingabe des Kommandos CONT möglich.

13.3 Fehlerbehandlungs-Routinen

In einigen voraussagbaren Fehlersituationen ist es nicht immer erwünscht, daß der Computer das laufende Programm abbricht. Ein solcher Fall ist denkbar, wenn z.B. bei einer Eingabe über den INPUT-Befehl der Benutzer einen falschen Datentyp verwendet und statt einer Zahl ein Wort eintippt. In dieser Situation wäre es nicht sehr sinnvoll, das Programm mit Ausgabe eines ERROR-Codes abzubrechen und unter Umständen erneut von Anfang an starten zu lassen. Der Hinweis auf den Irrtum und die Bitte um erneute Eingabe wäre wesentlich anwenderfreundlicher. Ebenso ließe sich in anderen Situationen wünschen, daß der Computer leichte Fehler selbst korrigiert. Um dieses alles realisieren zu können, steht die Anweisung ON ERROR GOTO zur Verfügung. Wird sie zu Beginn eines Programmes aktiviert, hat jeder auftretende Fehler nicht mehr einen Programmabbruch und die Anzeige eines ERROR-Codes zur Folge, sondern einen Sprung an die mit dieser Anweisung vereinbarte Zeile. Dort muß eine sogenannte Fehlerbehandlungs-Routine beginnen, die mit Hilfe der Systemvariablen ERN und ERL die Art und den Ort des Fehlers feststellen und daraus über die weiteren zu ergreifenden Maßnahmen entscheiden kann. Mit RESUME kann nach der Behandlung des Fehlers in den normalen Programmablauf wieder zurückgekehrt werden.

14 ERKLÄRUNG DER BASIC-BEFEHLE

14.1 Einleitung

Auf den folgenden Seiten möchten wir Ihnen nun den gewaltigen Befehlsvorrat des PC-1600 aufzeigen und erklären, welchem Zweck jeder einzelne Befehl dient und in welcher Weise er anzuwenden ist. Die Befehle sind dazu in alphabetischer Reihenfolge aufgelistet und mit Querverweisen versehen, so daß Sie dieses Kapitel auch als Nachschlagewerk verwenden können.

Einige der dort aufgeführten Befehle sind im MODE 0 (der eigentlichen Betriebsart des PC-1600) nicht von besonderem Nutzen und sollten vom Anwender besser durch geeignetere Befehle ersetzt werden. Der PC-1600 versteht sie nur deshalb, damit er im MODE 1 Kompatibilität zu seinem Vorgänger, dem PC-1500, gewährleisten kann.

14.2 Begriffsdefinitionen

Da ein Computer nicht von alleine tätig wird, muß man ihn erst dazu anweisen, anstehende Probleme oder Aufgaben zu lösen bzw. zu bearbeiten. Hierzu bedarf es eines geeigneten Verständigungsmittels, also einer Sprache. Diese Sprache ist in der Regel eine Programmiersprache, so daß sich alle Arbeitsgänge schrittweise als Einzelanweisungen formulieren, programmieren und bei Bedarf aktivieren lassen. Beim PC-1600 ist diese Programmiersprache eine SHARP-spezifische BASIC-Version, die sich, wie jede andere Sprache auch, aus einzelnen Wörtern zusammensetzt. Diese Wörter, die praktisch den Schlüssel zur Verständigung bilden, nennt man oftmals auch SCHLÜSSELWÖRTER. Sie bilden den Vorrat an Befehlen, mit denen man dem Computer Anweisungen erteilen kann. Aus diesem Grunde wollen wir künftig diese Schlüsselwörter vorwiegend als BEFEHLSWÖRTER oder schlicht BEFEHLE bezeichnen. Diese Befehle unterteilt man in der einschlägigen Fachliteratur meistenteils streng in Anweisungen und Kommandos. Unter Kommandos sind dabei vorwiegend jene Befehle gemeint, die in der Regel nur außerhalb eines Programmes anwendbar sind. Mit Anweisungen bezeichnet man die Befehle, die innerhalb eines Programmes akzeptiert werden. Diese Einteilung erweist sich als problematisch, da die Trennung nicht immer eindeutig zu ziehen ist und einige Befehle sowohl zu der einen als auch der anderen Kategorie gezählt werden können. Gerade beim SHARP-BASIC sind diese Grenzen noch weniger klar zu ziehen, so daß wir mit folgenden Begriffsdefinitionen arbeiten:

BEFEHL — Ein Befehl ist ein BASIC-Wort, das den Computer auf eine bestimmte durchzuführende Tätigkeit hinweist. Die Tätigkeit kann durch den Befehl selbst vollständig beschrieben sein. Bedarf der Befehl des Zusatzes von Parametern oder Optionen, so kann er nur als Anweisung verwendet werden.

ANWEISUNG — Eine Anweisung ist eine komplette Tätigkeitsaufforderung, bestehend aus dem Befehlswort und allen notwendigen oder optionalen Parametern.

Damit der Computer die an ihn herangetragenen Anweisungen auch verstehen kann, müssen gewisse Regeln eingehalten werden, die die Angabe von Parametern und Optionen usw. beschreiben. Diese Regeln bezeichnet man als SYNTAX. Je nach Befehl kann diese Syntax mehr oder weniger komplex sein. Es ist klar, daß sich komplizierte Syntaxregeln nicht einfach erklären lassen. Deshalb haben wir uns dazu entschieden, Ihnen diese Regeln durch Bilder, sogenannte SYNTAX-Diagramme, zu veranschaulichen. Anhand dieser Diagramme lassen sich die Regeln jederzeit schnell ablesen, ohne einen Wust von Erklärungen lesen zu müssen. Zudem stellen sie sicher, daß auch Zusammenhänge vom Anwender erkannt werden, die unter Umständen bei den Erklärungen der Befehle außer acht gelassen worden sind. Falls für Sie die Syntax-Diagramme etwas völlig neues sind, möchten wir Sie bitten, den Anhang K zu studieren, bevor Sie sich die Erklärungen der BASIC-Befehle zu Gemüte führen.

14.3 Befehls-Kürzel

Fast alle BASIC-Befehle des PC-1600 können durch Abkürzungen ersetzt werden. Dazu brauchen nur die wichtigsten Buchstaben, die den jeweiligen Befehl eindeutig charakterisieren mit einem nachfolgenden Punkt eingegeben zu werden. So sind beispielsweise für das Befehlswort RADIAN folgende Eingaben zulässig:

```
RADIAN
RADIA.
RADI.
RAD.
```

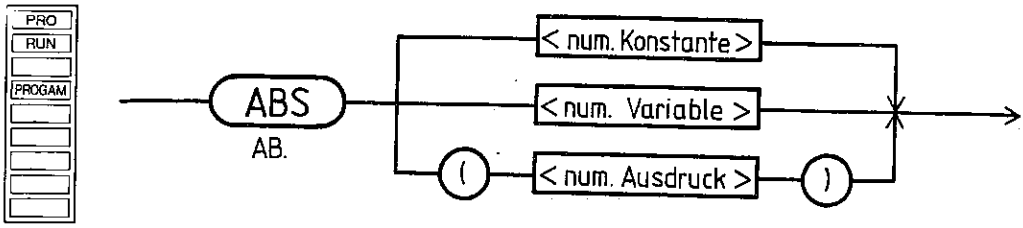
Die jeweils kürzeste Form eines jeden Befehles finden Sie in den Syntax-Diagrammen unterhalb des vollständig ausgeschriebenen Befehlswortes.

14.4 Anwendungssymbolik

Wie schon unter 14.2 erwähnt, ist es nicht immer einfach, durch eine Kategorisierung zu beschreiben, ob ein Befehl nun nur auf Kommando-Ebene oder innerhalb eines Programmes anwendbar ist, also ob ein Befehl nur im RUN-Modus oder im PRO-Modus akzeptiert wird. Darüberhinaus sind einige Befehle auch nur dann wirklich nutzbar, wenn spezielle Peripheriegeräte oder andere Optionen am Computer angeschlossen sind. Sind diese Randbedingungen nicht erfüllt, so kommt es zu einer Fehlermeldung, also der Ausgabe eines ERROR-Codes. Damit Sie diese Randbedingungen nicht mühsam aus dem Erklärungstext zusammensuchen müssen, verwenden wir die nachstehend erläuterte Symbolik.

Diese Symbole sollen darüber Auskunft geben, in welchem Modus ein Befehl anwendbar ist und welches Zubehör dabei am Computer angeschlossen sein muß.

Symbol	Bedeutung
PRO	Der Befehl ist direkt im PRO-Modus ausführbar.
RUN	Der Befehl ist direkt im RUN-Modus ausführbar
RESERVE	Nutzbar zur Funktionstasten-Belegung im RESERVE-Modus.
PROGAM	Dieser Befehl kann programmiert werden.
	Cassetten-Befehl Es sollte ein Cassetten-Recorder angeschlossen sein. (CE-1600P oder CE-150 erforderlich !)
	Disketten-Befehl Das Disketten-Laufwerk CE-1600F muß angeschlossen sein.
RAM	RAM-Disk-Befehl. Ein Programm-Modul (CE-1600M, CE-161) muß in einem der beiden Modulfächer eingesetzt sein.
	Druck-Befehl Es muß einer der beiden Drucker CE-1600P oder CE-150 angeschlossen sein.
	Kommunikations-Kommando. Dieser Befehl dient der Bedienung der seriellen Ports. Es sollte daher an dem aktiven Interface ein geeignetes Peripheriegerät angeschlossen sein.



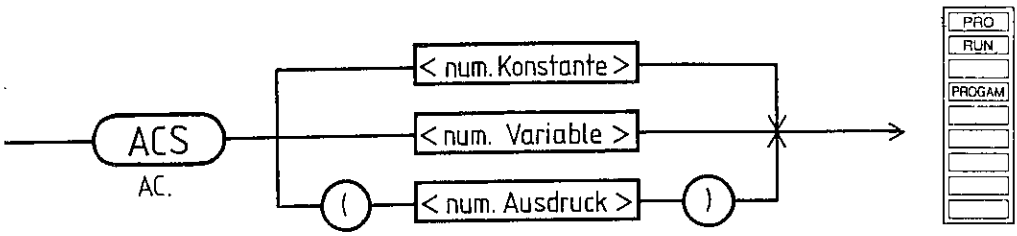
WIRKUNG : Die Funktion ABS liefert den Absolutbetrag eines numerischen Ausdruckes.

BEISPIEL : Schalten Sie bitte mit Hilfe der MODE-Taste den PC-1600 in den PRO-Modus und geben Sie folgendes kleine BASIC-Programm ein. Vermeiden Sie dabei aber die Eingabe der Doppelpunkte, die direkt hinter den Zeilennummern angegeben sind. Diese werden vom Computer nämlich automatisch eingefügt, sobald Sie die ENTER-Taste betätigen und damit dem PC-1600 zu verstehen geben, daß er die Programmzeile in seinen Speicher aufnehmen soll. Der Doppelpunkt ist also eine Rückmeldung bzw. eine Bestätigung dafür, daß die Programmierung der Zeile erfolgt ist. Alle anderen im Programm vorkommenden Doppelpunkte sind jedoch wie gezeigt einzutippen.

```
10:PRINT "ZAHL  :";
20:FOR X=-2 TO 2:PRINT X;" ";:NEXT X
30:PRINT
40:PRINT "BETRAG:";
50:FOR X=-2 TO 2:PRINT ABS(X);" ";:NEXT X
60:PRINT
70:END
```

Schalten Sie nun den Computer in den RUN-Modus und starten Sie das Programm mit dem Befehl RUN. (Vergessen Sie nicht die Betätigung der Taste ENTER .) Auf dem Display erscheint dann:

```
>
RUN
ZAHL  :-2 -1  0  1  2
BETRAG: 2  1  0  1  2
>
```



Siehe auch : ASN, ATN, COS

WIRKUNG : Der Befehl ACS(X) liefert einen zum Argument X gehörenden Wert der Arcus-Cosinus-Funktion.

HINWEISE : Da die Arcus-Cosinus-Funktion die Umkehrung der Cosinus-Funktion ist, stellt der gelieferte Wert folglich einen Winkel dar. In Abhängigkeit von dem derzeit gültigen Winkelmodus (DEGREE, GRAD, oder RADIAN) ist das Ergebnis somit entweder in Altgrad, Neugrad oder im Bogenmaß zu werten. Das zulässige Funktionsargument X ist auf den Wertebereich $-1 \leq X \leq 1$ beschränkt. Der gelieferte Funktionswert liegt stets in folgenden Hauptwertebereichen:

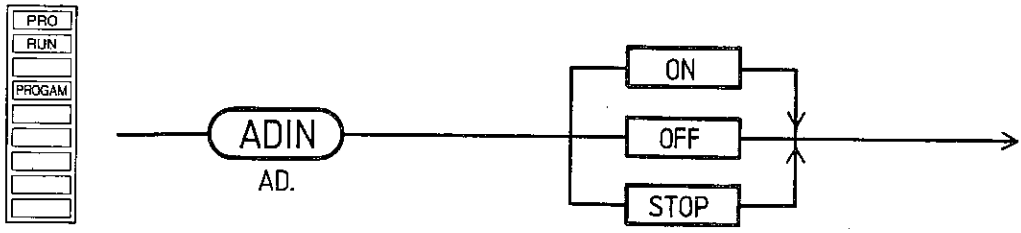
DEG-Modus :	0°.... 180°
RAD-Modus :	0 .... π
GRAD-Modus :	0 .... 200 gon

BEISPIEL :

```
10:DEGREE
20:PRINT "arccos(0.5)=";ACS(.5);" Grad"
30:PRINT "arccos(0)  =";ACS(0);" Grad"
40:END

>
RUN
arccos(0.5)= 60 Grad
arccos(0)  = 90 Grad
>
```

ADIN ON/OFF/STOP



WIRKUNG : Der Befehl ADIN erlaubt oder verhindert die Annahme von analogen Interrupt-Anforderungen.

HINWEISE : Eine analoge Interrupt-Anforderung liegt vor, wenn das am Analog-Eingang anstehende Signal innerhalb eines vereinbarten Pegelbereiches liegt (siehe hierzu: ON ADIN GOSUB).

ADIN IN erlaubt die Annahme eines analogen Interrupts. Mit der Anweisung ON ADIN GOSUB kann im Interrupt-Fall dann in eine Interrupt-Behandlungsroutine verzweigt werden.

ADIN OFF verhindert die Annahme von analogen Interrupts.

ADIN STOP schaltet ebenfalls die Annahme analoger Interrupts aus, registriert jedoch die jeweils letzte Anforderung in einem Speicher. Sobald mit der Anweisung ADIN ON dann die Annahme von analogen Interrupts freigegeben wird, erfolgt die unmittelbare Verzweigung in die Interrupt-Routine. STOP ist der standardmäßig vom Sytem angenommene ADIN-Parameter.

Siehe auch : ADIN ON/OFF/STOP, ON ADIN GOSUB

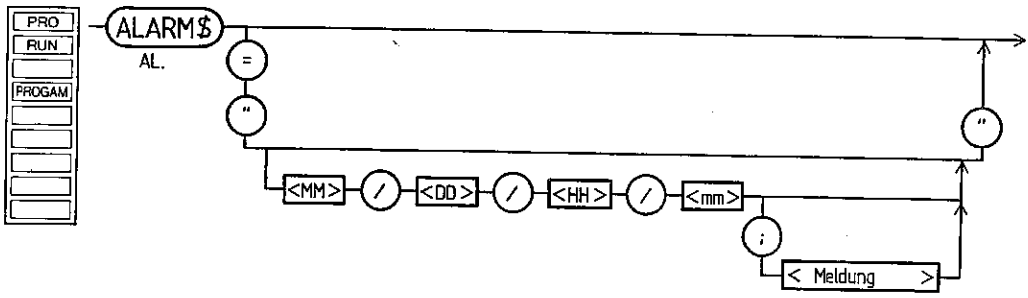
WIRKUNG : AIN liefert einen Wert, der dem Spannungspegel am analogen Eingang entspricht.

HINWEIS : Der in der speziellen Variablen AIN enthaltene Integer-Wert kann zwischen 0 und 255 liegen.

Dieser Werte-Bereich deckt dabei alle Spannungen von 0 bis 2.495 V ab. Höhere Spannungen liefern den Integer-Wert 255.

BEISPIEL :
```
PRINT AIN
43
>
```

ALARM$



WIRKUNG : Mit ALARM$ läßt sich eine Alarmzeit und eine Alarmmeldung vereinbaren.

HINWEISE : **ALARM$="MM/DD/HH/mm"**

.... setzt die Alarmzeit, wobei die aufgeführten Platzhalter für die tatsächlichen Werte folgende Bedeutung haben:

Platzhalter	Bedeutung	zulässige Werte
MM	Monat (month)	01 12
DD	Tag (day)	01 31
HH	Stunde (hour)	00 23
mm	Minute (minute)	00 59

Anstelle aktueller Werte können auch "wildcards" in Form von Fragezeichen für den Monat und den Tag verwendet werden. So setzt beispielsweise die Angabe "??/??/13/30" die Alarmzeit auf 13:30 des heutigen Tages.

Wird die mit dieser Anweisung vereinbarte Uhrzeit erreicht, gibt der Computer für die Dauer einer Sekunde wiederholte Piepstöne von sich.

Zusätzlich kann mit der Anweisung ALARM$ eine **<Meldung>** vereinbart werden, die bei erreichter Alarmzeit zusammen mit dieser Zeitangabe auf dem dem Display angezeigt wird. Diese <Meldung> darf aus maximal 26 Zeichen bestehen.

Die <Meldung> überschreibt die Funktionstastenbelegung der Ebene II.

ALARM$=""

..... löscht sowohl die gesetzte Alarmzeit als auch eine eventuell vereinbarte Alarmmeldung.

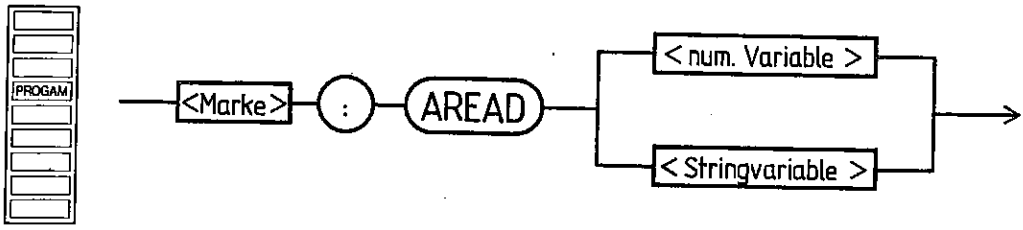
Das Befehlswort ALARM$ kann darüberhinaus als String-Variable verwendet werden, um z.B. die Vereinbarungen von Alarmzeit und Alarmmeldung sichtbar zu machen.

BEISPIEL : Angenommen heute sei der 12. Oktober und Sie hätten am nächsten Tag um 14:30h einen wichtigen Termin. Mit der Anweisung

```
ALARM$="10/13/13/45;TERMIN UM 14H30"
```

können Sie sich dann vom PC-1600 rechtzeitig an diesen erinnern lassen. Diese Vorsichtsmaßnahme nützt Ihnen allerdings nur, wenn der Computer zum Alarmzeitpunkt eingeschaltet ist. Im ausgeschalteten Zustand ertönt nur ein Piepssignal, es erscheint aber nicht die Meldung auf dem Display, da hierzu der Computer eingeschaltet sein müßte. Falls sich das Gerät automatisch zu einer bestimmten Zeit einschalten soll, müssen Sie den WAKE$-Befehl verwenden.

AREAD



Siehe auch : RUN

WIRKUNG : Mit AREAD kann ein in der Anzeige befindlicher String oder numerischer Wert mit dem Start des Programmes in eine bereitgestellte Variable eingelesen werden.

HINWEISE : Diese Art der Datenübergabe an ein Programm geht nur unter folgenden Voraussetzungen:

a) Das Befehlswort AREAD muß in der ersten Programm-Zeile stehen.

b) Es hat ihm unmittelbar eine Markierung vorauszugehen. Diese darf nur aus einem der nachstehend genannten Buchstaben bestehen:

A,S,D,F,G,H,J,K,L,Z,X,C,V,B,N,M

c) Das Programm muß über die DEF-Taste in Verbindung mit der gewählten Marke in Betrieb genommen werden.

d) Der in der Anzeige befindliche Wert und die mit AREAD bereitgestellte Variable müssen typenmäßig übereinstimmen.

Weist die Anzeige einen numerischen Wert auf, so wird dieser mit bis zu 10 Mantissen- und bis zu 2 Exponenten-Ziffern eingelesen, wobei sowohl die Mantisse als auch der Exponent vorzeichenbehaftet sein darf.

Bei Strings hängt die Anzahl der eingelesenen Zeichen von der über den DIM-Befehl definierten Stringlänge der Variablen ab. Eine nicht DIMensionierte Variable kann standardmäßig bis zu 16 Zeichen aufnehmen.

Steht bei Ausführung des AREAD-Befehles nur das Bereitschaftszeichen > in der Anzeige, so wird einer numerischen Variable der Wert 0 und einer Stringvariable ein Nullstring (ASCII-Code &00) zugewiesen.

BEISPIEL :

```
10:"F":AREAD K$
20:PRINT "** ";K$
```

Start bei leerem Display:

```
CLS       Display löschen
DEF  F    Programm starten
```

Anzeige:

```
**
>
```

Start bei beschriebenem Display:

```
HALLO     Text eingeben
DEF  F    Programm starten
```

Anzeige:

```
** HALLO
>
```

WIRKUNG : Ein im Arbeitsspeicher befindliches und mit dem Befehl ARUN versehenes Programm wird beim Einschalten des Computers automatisch gestartet.

HINWEISE : Dazu müssen allerdings folgende Bedingungen erfüllt sein:

a) ARUN muß sich in der ersten Programmzeile befinden, und zwar unmittelbar hinter der Zeilennummer. Es darf diesem Befehl also weder ein anderer Befehl noch eine Marke vorausgehen.

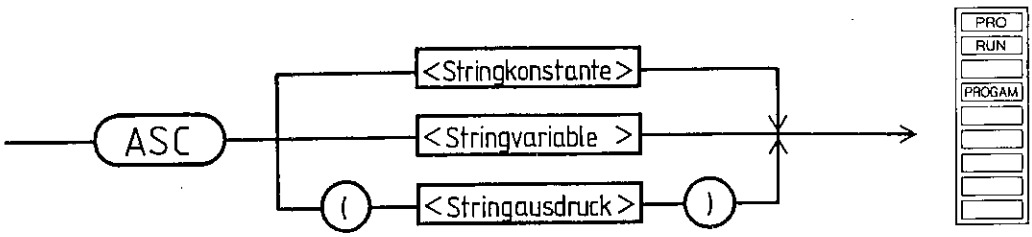
b) Der Computer muß im RUN-Modus ausgeschaltet worden sein, damit er sich beim Einschalten ebenfalls in diesem Modus befindet, da nur in dieser Betriebsart Programme ablauffähig sind.

Werden bei ausgeschaltetem Computer irgendwelche Optionen (z.B. Drucker, RAM-Module, Cassetten-Recorder, Diskettenlaufwerk usw.) angeschlossen oder gewechselt, so kann es bei der Einschaltung des Computers unter Umständen zur Ausgabe eines ERROR-Codes kommen. In diesem Falle unterbleibt natürlich ein automatischer Programm-Start und das Programm muß über RUN manuell in Betrieb genommen werden.

Ein mit ARUN gestartetes Programm löscht keine Variablen. Sollte eine derartige Löschung jedoch erwünscht sein, so ist im Programm der Befehl CLEAR an geeigneter Stelle einzufügen.

BEISPIEL :

```
10:ARUN:CLS
20:PRINT"GUTEN TAG !"
30:PRINT"ES IST JETZT ";TIME$;" UHR."
40:PRINT"ES SIND ";MEM;" BYTES FREI."
50:END
```



Siehe auch : CHR$

WIRKUNG : Die Funktion ASC(X$) liefert den ASCII-Code des Argumentes X$, welches ein String aus einem oder mehreren Zeichen sein kann.

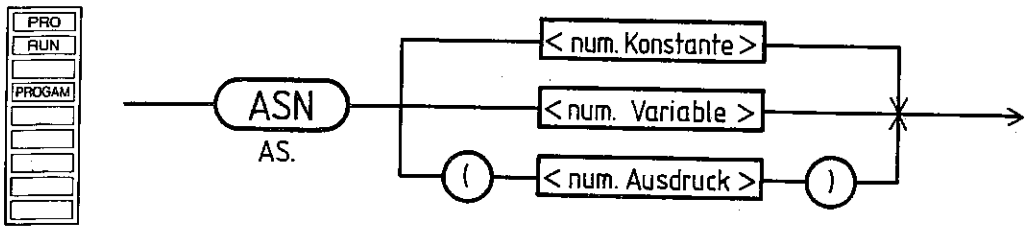
HINWEISE : Wird als Funktionsargument ein String genommen, der aus mehr als einem Zeichen besteht, so wird der ASCII-Code seines ersten Zeichens geliefert.

Der Zusammenhang zwischen dem gelieferten Code und dem zugehörigen Zeichen ist aus dem Anhang C ersichtlich.

BEISPIEL :

```
10:WAIT 0:CLS
20:PRINT "BITTE EIN ZEICHEN ODER"
30:INPUT "EINEN STRING EINGEBEN:",S$
40:WAIT 100
50:PRINT "DER ASCII-CODE LAUTET:";ASC(S$)
60:END

>
RUN
BITTE EIN ZEICHEN ODER
EINEN STRING EINGEBEN:
SHARP
DER ASCII-CODE LAUTET: 83
>
RUN
BITTE EIN ZEICHEN ODER
EINEN STRING EINGEBEN:
*
DER ASCII-CODE LAUTET: 42
```



Siehe auch : ACS, ATN, SIN

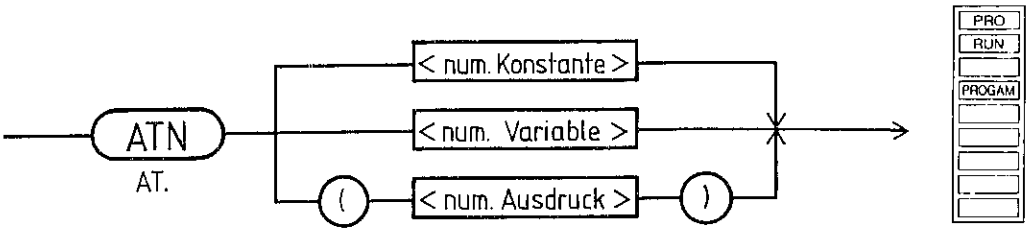
WIRKUNG : Der Befehl ACS(X) liefert einen zum Argument X gehörenden Wert der Arcus-Sinus-Funktion.

HINWEISE : Da die Arcus-Sinus-Funktion die Umkehrung der Sinus-Funktion ist, stellt der gelieferte Wert folglich einen Winkel dar. In Abhängigkeit von dem derzeit gültigen Winkelmodus (DEGREE, GRAD, oder RADIAN) ist das Ergebnis somit entweder in Altgrad, Neugrad oder im Bogenmaß zu werten. Das zulässige Funktionsargument X ist auf den Wertebereich $-1 \leq X \leq 1$ beschränkt. Der gelieferte Funktionswert liegt stets im Hauptwertebereich. Hierbei gilt:

DEG-Modus :	$-90° \ldots 90°$
RAD-Modus :	$-\pi/2 \ldots \pi/2$
GRAD-Modus :	(-100 .... 100) gon

BEISPIEL :

```
10:DEGREE
20:GOSUB 100
30:FOR DX=-10 TO 10
40:X=DX/10
50:F=ASN(X):Z=Z+1
60:IF Z=3 THEN GOSUB 100
70:PAUSE "    ";STR$(X),F
80:NEXT DX
90:END
100:CLS:PRINT "ARGUMENT","ARCUS-SINUS"
110:Z=0:RETURN
```



Siehe auch: ACS, ASN, TAN

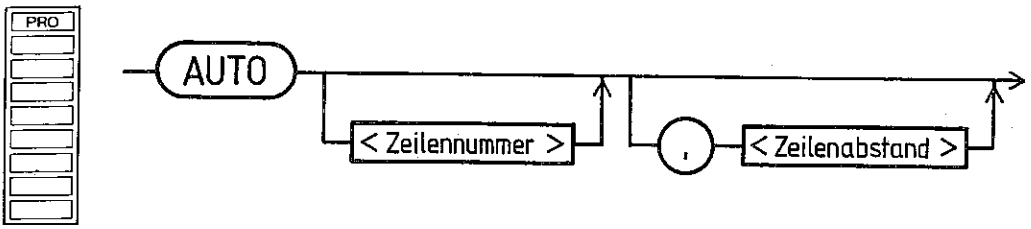
WIRKUNG : Der Befehl ATN(X) liefert einen zum Argument X gehörenden Wert der Arcus-Tangens-Funktion.

HINWEISE : Da die Arcus-Tangens-Funktion die Umkehrung der Tangens-Funktion ist, stellt der gelieferte Wert folglich einen Winkel dar. In Abhängigkeit von dem derzeit gültigen Winkelmodus (DEGREE, GRAD, oder RADIAN) ist das Ergebnis somit entweder in Altgrad, Neugrad oder im Bogenmaß zu verstehen. Das Funktionsargument X unterliegt keiner wertmäßigen Beschränkung. Der Funktionswert wird je nach Winkelmodus innerhalb folgender Hauptwertebereiche geliefert:

DEG-Modus :	-90° .... 90°
RAD-Modus :	$-\pi/2$ .... $\pi/2$
GRAD-Modus :	(-100 .... 100) gon

BEISPIEL :

```
10:DEGREE
20:GOSUB 100
30:FOR DX=0 TO 100
40:X=DX*.1
50:F=ATN(X):Z=Z+1
60:IF Z=3 THEN GOSUB 100
70:PAUSE "    ";STR$(X),F
80:NEXT DX
90:END
100:CLS:PRINT "ARGUMENT","ARCUS-TANGENS"
110:Z=0:RETURN
```



Siehe auch : RENUM

WIRKUNG : Mit dem Kommando AUTO kann zur Erleichterung des Programmierens im PRO-Modus eine automatische Zeilennumerierung vorgenommen werden.

HINWEISE : Nach Aktivierung von AUTO erscheint die erste generierte Zeilennummer in der Anzeige mit einem nachgestellten Cursor. Nun kann der gewünschte Zeileninhalt eingegeben werden. Schließt man die Eingabe dann durch Betätigung der ENTER-Taste ab, so wird in der folgenden Zeile die nächste Zeilennummer generiert und so fort.

Ergibt sich bei der Generation der Zeilennummern die Nummer einer bereits existierenden Zeile, so wird diese Zeile angezeigt.

Die erzeugten Zeilennummern hängen davon ab, ob AUTO mit oder ohne Parameter versehen wird und welche Werte für diese gewählt werden:

AUTO

Wird das Kommando AUTO ohne Parameter angegeben, so beginnt die Numerierung mit der Zeile 10 und setzt sich im Zehnerabstand, also mit den Zeilen 20, 30 usw. fort.

AUTO <Zeilennummer>

Ist dem Befehlswort AUTO eine einzelne Integerzahl beigefügt, gilt diese als die erste Zeilennummer. Alle weiteren Zeilennummern folgen im Zehnerabstand.

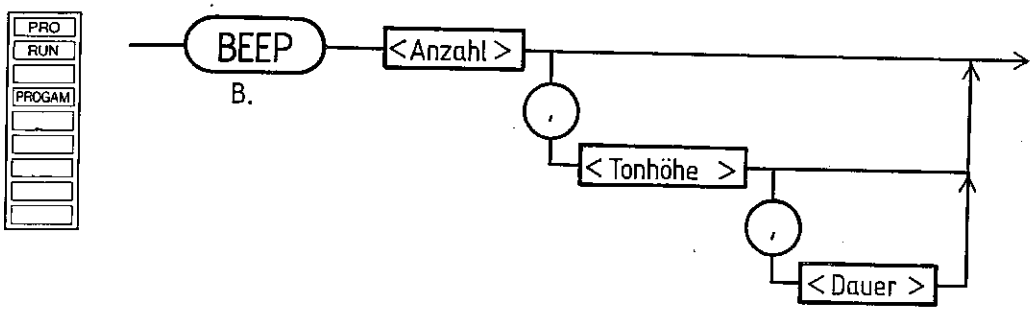
AUTO <Zeilennummer>,<Zeilenabstand>

Mit einem weiteren Parameter kann die Schrittweite der Zeilennumerierung bestimmt werden. Die beiden Parameter sind durch ein Komma voneinander zu trennen.

Die automatische Zeilennumerierung läßt sich mit Betätigung der BREAK- oder CL-Taste aufheben. Es kann aber auch die ENTER-Taste zu diesem Zweck verwendet werden, wenn man diese gleich nach dem Erscheinen einer neuen Zeilennummer betätigt, ohne zuvor etwas in die Zeile hineinzuschreiben.

BEISPIELE:

Anweisung	Generierte Zeilennummern
AUTO	10,20,30,40,.....
AUTO 100	100,110,120,.....
AUTO 400,20	400,420,440,.....



Siehe auch : BEEP ON/OFF

WIRKUNG : BEEP erzeugt eine Anzahl von Tönen spezifizierte Tonhöhe und Dauer.

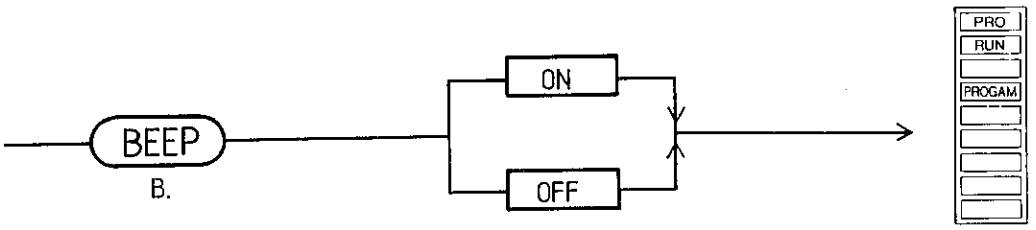
<Anzahl> bestimmt, wie oft der standardmäßige oder aber näher spezifizierte Ton vom Computer erzeugt werden soll. Die zulässigen Werte sind: 0 ... 65535.

<Tonhöhe> legt die Frequenz des Tones fest und darf durch einen Integer-Wert von 0 bis 255 vertreten sein. Es lassen sich Frequenzen von 230 Hz bis 7 kHz erzeugen, wobei Frequenz und Wert des Parameters in einem umgekehrt "proportionalen" Verhältnis stehen. Je höher der Wert, desto niedriger die Frequenz:

0 bedeutet ca. 7 kHz
255 bedeutet ca. 230 Hz

Fehlt dieser Parameter, wird als Standard eine Frequenz von 4 kHz geliefert.

<Dauer> bestimmt die Dauer eines Tones. Sie ist abhängig von der Frequenz, also dem Wert von <Tonhöhe>. Je tiefer die Frequenz, umso länger die Dauer. Der Wert darf im Bereich 0...65279 liegen und wird bei fehlendem Parameter zu 160 angenommen.



Siehe auch : BEEP

WIRKUNG : BEEP ON/OFF schaltet die Wirksamkeit des BEEP-befehles ein bzw. aus.

HINWEISE : Die Kommandos BEEP ON/OFF haben ebenfalls einen Einfluß auf die Wiedergabe des Kontroll-Tones, der beim Einladen eines Programmes von einer Cassette geliefert wird.

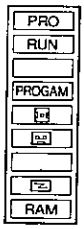
BEEP ON erlaubt das Wirksamwerden einer BEEP-Anweisung. Beim Laden von Programmen bzw. Daten von Cassette werden diese über den eingebauten Lautsprecher hörbar gemacht, um so den Ladevorgang kontrollieren zu können.

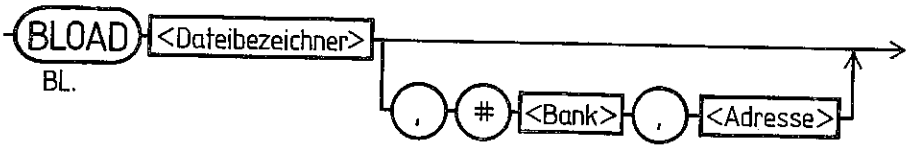
BEEP OFF schaltet den Lautsprecher ab. Es wird dann weder eine BEEP-Anweisung wirksam noch ein Lade-Kontrollton hörbar.

BEISPIEL :

```
10:BEEP 4
20:BEEP OFF
30:BEEP 4
40:BEEP ON
50:END
```

Zeile 10 erzeugt vier gleichfrequente Töne
Zeile 20 schaltet den Lautsprecher ab
Zeile 30 erzeugt wegen Zeile 20 keinen Ton
Zeile 40 schaltet den Lautsprecher wieder ein
Zeile 50 beendet das Programm





Siehe auch : BSAVE, CLOAD, NEW, SET

WIRKUNG : BLOAD lädt ein Maschinensprache-Programm von einer Diskette oder einem RAM-Disk-Modul.

HINWEISE : Der Parameter **<Dateibezeichner>** bestimmt die Option von der das Programm zu holen und unter welchem Namen und Gruppenkennung (extension) es dort abgelegt ist. Der <Dateibezeichner> hat das Format:

<Datenquelle:Dateiname.Extension>

Folgende Datenquellen können adressiert werden:

S1: oder S2: RAM-Modul des Faches S1 oder S2
X: oder Y: Diskette
COM1: RS-232C-Schnittstelle
COM2: Optoelektronische Schnittstelle

<Bank> spezifiziert die Speicherbank, in die das Maschinensprache-Programm geladen werden soll. Es sind die Banknummern 0 bis 7 möglich.

<Adresse> bestimmt die Adresse innerhalb der gültigen Speicherbank, ab der die Ablage des Programmes erfolgen soll. Der zulässige Wertebereich lautet: 0 ... 65535.

Werden die beiden optionalen Parameter <Bank> und <Adresse> nicht angegeben, so legt sich das Programm in genau die Speicherbank ab derjenigen Anfangsadresse ab, wo es sich vor der Sicherung zuvor im Arbeitsspeicher befunden hat.

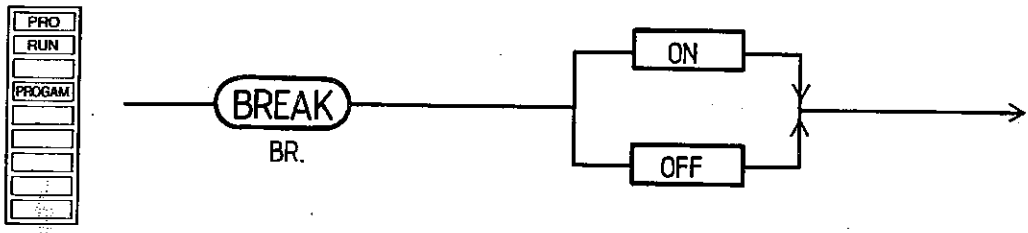
Wurde das Programm mit dem BSAVE-Kommando unter Angabe einer Auto-Startadresse gesichert, so wird es genau ab dieser Adresse in den Speicher geladen und nach Beendigung des Ladevorganges automatisch ausgeführt.

Ist das auf Diskette befindliche Programm mit der I-Option des SET-Befehles geschützt, bleibt diese Option unbeachtet.

Schauen Sie bitte in den Anhang D, um weitere Informationen über die Speicherbänke und Adreßbereiche zu erhalten.

BEISPIEL : BLOAD "X:RXOUT"

BREAK ON/OFF



Siehe auch : CONT

WIRKUNG : Erlaubt oder verbietet das Wirksamwerden der Betätigung der BREAK-Taste.

HINWEISE : BREAK OFF schaltet die Funktion der BREAK-Taste ab. Ein laufendes Programm kann dann durch Betätigung dieser Taste nicht unterbrochen werden.

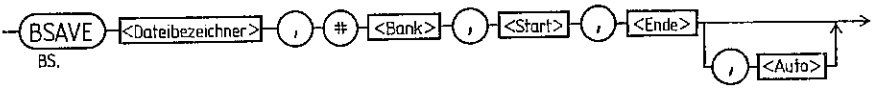
BREAK ON schaltet die Funktion der BREAK-Taste ein. Eine Betätigung der BREAK-Taste bricht dann ein laufendes Programm ab und gibt dabei folgende Meldung aus:

BREAK IN <Zeilennr.>

Mit dem CONT-Kommando kann der Ablauf des Programmes fortgesetzt werden.

Eine gute Idee für den Einsatz der Anweisungen BREAK ON und BREAK OFF ist, diese unmittelbar vor und hinter einem Programmteil einzubinden, das auf keinen Fall vom Anwender unterbrochen werden darf, weil z.B. gerade wichtige Daten übertragen oder eine Grafik gedruckt wird.

Falls durch die Anwendung dieser Anweisungen ein Programm nicht aus einer Endlosschleife befreit werden kann, bleibt nur der Weg, die RESET-Taste zu bedienen. Die Anweisung BREAK OFF sollte also nur dann eingesetzt werden, wenn ein Programm einen fehlerfreien Ablauf gewährleistet und das Auftreten solcher Endlosschleifen ausgeschlossen werden kann.



Siehe auch : BLOAD, CSAVE M

WIRKUNG : Sichert ein Maschinensprache-Programm auf einer Diskette oder einem RAM-Disk-Modul.

HINWEISE : Der Parameter **<Dateibezeichner>** bestimmt, auf welchem Medium das Programm zu sichern und unter welchem Namen und welcher Extension es dort abzulegen ist. Er hat das Format:

<Datenquelle:Dateiname.Extension>

Im Gegensatz zum SAVE-Kommando muß bei diesem <Dateibezeichner> die Extension unbedingt angegeben werden.

Folgende Datenquellen sind adressierbar:

S1: oder S2:	RAM-Modul des Faches S1 oder S2
X: oder Y:	Diskette
COM1:	RS-232C-Interface
COM2:	Optoelektronisches Interface

<Bank> bestimmt die Speicherbank, auf der das Maschinensprache-Programm vorzufinden ist. Es sind die Banknummern 0 bis 7 möglich.

<Start> bestimmt die Adresse innerhalb der gültigen Speicherbank, ab der das Programm beginnt. Für Adresse sind die Werte 0 ... 65535 zulässig.

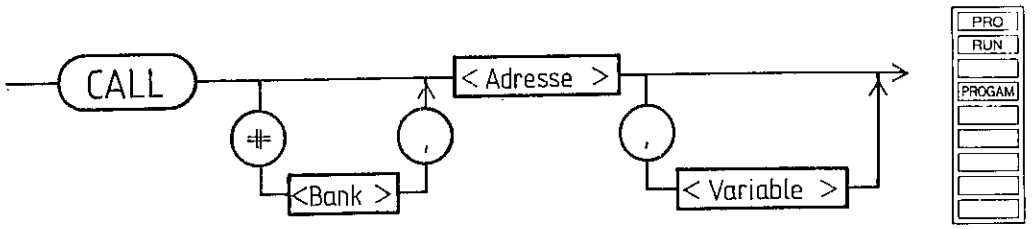
<Ende> bestimmt die höherwertige Adresse innerhalb der gültigen Speicherbank, mit der das Programm endet.

<Auto> bestimmt, ab welcher Adresse das Programm automatisch gestartet werden soll, wenn man es mit BLOAD wieder an seinen Platz zurücklädt. Fehlt dieser optionale Parameter, so wird er auf den Standard-Wert &FFFF gesetzt und damit ein Auto-Start verhindert.

Über die Aufteilung des Speichers in Bänke und deren Adreßbereiche können Sie sich anhand des Anhanges D informieren.

BEISPIEL : BSAVE "S1:SORT",#1,&8000,&8AFF

Diese Anweisung sichert ein auf Speicherbank 1 befindliches Maschinen-Programm von der Adresse &8000 beginnend bis einschließlich der Adresse &8AFF auf dem im Modulfach S1 befindlichen RAM-Modul unter dem Namen SORT.



Siehe auch : NEW, POKE, XPOKE

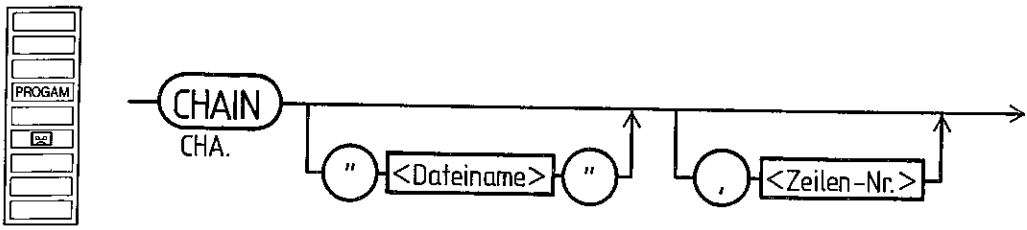
WIRKUNG : Mit CALL kann von einem BASIC-Programm oder der RUN-Ebene aus, ein Maschinensprache-Programm gestartet und anschließend in die aufrufende Ebene zurückgekehrt werden. Dabei ist die Übergabe eines einzelnen Variablen-Wertes möglich.

HINWEISE : **<Bank>** bestimmt die Speicherbank aus dem Bereich 0...7, in der das Maschinenspracheprogramm gespeichert ist. Wird dieser Parameter nicht angegeben, gilt Speicherbank 0.

<Adresse> nennt die Anfangsadresse innerhalb der gültigen Speicherbank, bei der das Programm beginnt. Es sind die Adressen von 0...65535 (&0...&FFFF) zulässig. Die Adreßangabe muß erfolgen und darf nicht weggelassen werden.

<Varibale> bestimmt die Variable, deren Wert an das Maschinensprache-Programm übergeben wird, und in die nach Beendigung des Programmes der aktualisierte Wert abzugeben ist. In diese Variable wird jedoch nur dann ein Wert zurückgegeben, wenn bei Rückkehr auf die BASIC-Ebene das Carry-Flag gesetzt ist. Bei Variablen numerischen Typs wird der Wert an das Registerpaar DE abgegeben. Da er vorzeichenbehaftet sein kann, muß er folglich im Bereich von -32768 bis 32767 liegen. Bei Stringvariablen nimmt dieses Registerpaar die Adresse auf, ab der der String im Speicher abgelegt ist. Register B führt den Wert der Stringlänge.

BEISPIEL : s. POKE



Siehe auch : CSAVE, MERGE

WIRKUNG : CHAIN lädt von einem BASIC-Programm aus ein auf Cassette befindliches anderes BASIC-Programm und startet es.

HINWEISE : **CHAIN**

lädt das erste auf Cassette auffindbare Programm in den Speicher und beginnt dessen Abarbeitung mit der ersten Zeile.

CHAIN <Zeilen-Nr.>

verhält sich wie CHAIN ohne Parameter, startet das Programm jedoch mit der angegebenen Zeile.

CHAIN <Dateiname>

sucht auf der Cassette nach dem Programm, das den spezifizierten Namen trägt, lädt dieses und startet es mit seiner ersten Zeile.

CHAIN <Dateiname>,<Zeilen-Nr.>

lädt das Programm angegebenen Namens und nimmt es ab der spezifizierten Zeile in Betrieb.

Die Anwendung des CHAIN-Befehles erlaubt, lange Programme, die nicht in den Speicher passen, als einzelne Teilprogramme zerlegt, Stück für Stück nacheinander automatisch zu laden und ablaufen zu lassen.

Sollte eines der zu ladenden Programme mittels PASS-Befehl geschützt sein, so generiert CHAIN die Ausgabe eines ERROR-Codes und stoppt damit den Programmablauf.

BEISPIEL :

```
>
RUN
```

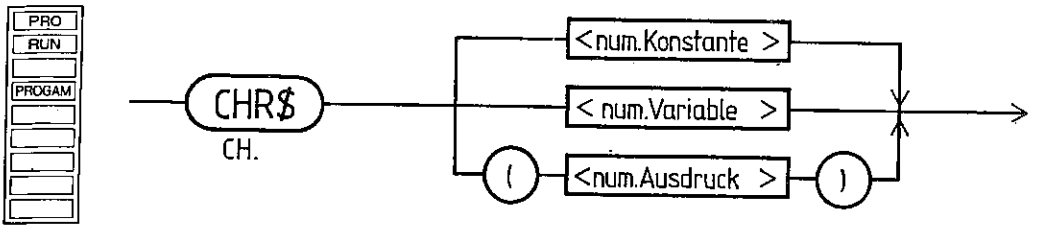
PROG1

```
10:REM PROGRAMM 1
20:
30:
:
:
400: CHAIN "PROG2"
```

PROG2

```
10:REM PROGRAMM 2
20:
30:
:
:
200:END
```

Mit dem RUN-Befehl wird das gerade im Speicher befindliche Programm PROG1 gestartet. Erreicht dieses Programm seine letzte Zeile, so sucht der Computer wegen des CHAIN-Befehles auf der Cassette nach dem Programm PROG2 und lädt es bei Vorhandensein in den Speicher, wobei das Programm PROG1 dabei überschrieben wird. Ist der Ladevorgang abgeschlossen, wird das neu in den Speicher geholte Programm mit seiner ersten Zeile gestartet, da der CHAIN-Befehl ohne Parameter <Zeilen-Nr.> in PROG1 angegeben ist.



Siehe auch : ASC

WIRKUNG : Die Funktion CHR$ liefert das Zeichen, dessen zugehöriger ASCII-Code als Funktionsargument angegeben ist.

HINWEISE : Das betreffende Argument kann entweder eine Konstante, eine Variable oder ein Ausdruck sein. Der numerische Wert dieser Größen muß aber in jedem Falle vom Typ Integer sein.

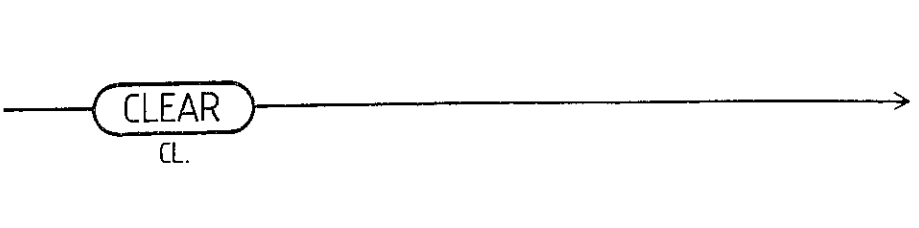
Mit dieser Funktion können Steuerzeichen, die nicht über die Tastatur zugänglich sind, an die verschiedensten Peripheriegeräte wie Drucker usw. oder aber an die seriellen Schnittstellen weitergeleitet werden.

In welcher Zuordnung die Zeichen zu den ASCII-Codes stehen, können Sie aus Anhang C ersehen.

BEISPIEL :

```
10:FOR X= 33 TO 126
20:PAUSE CHR$(X);
30 NEXT X
40:END
```

Dieses Programm zeigt alle darstellbaren Zeichen des standardmäßigen ASCII-Bereiches (Codes: &21 bis $7E) auf dem Display an. Das erste Zeichen ist hierbei das Ausrufezeichen !, das letzte die sogenannte Tilde ~.



Siehe auch : DIM, ERASE, TITLE

WIRKUNG : CLEAR löscht sämtliche im Speicher befindlichen Variablen. Dies gilt auch für die reservierten, sprich Standard-Variablen.

HINWEISE : Die numerischen Standardvariablen A bis Z bzw. @(1) bis @(26) werden dabei mit dem Wert 0 belegt und den Stringvariablen A$ bis Z$ bzw. @$(1) bis @$(26) ein Nullstring (ASCII-Code 0) zugewiesen.

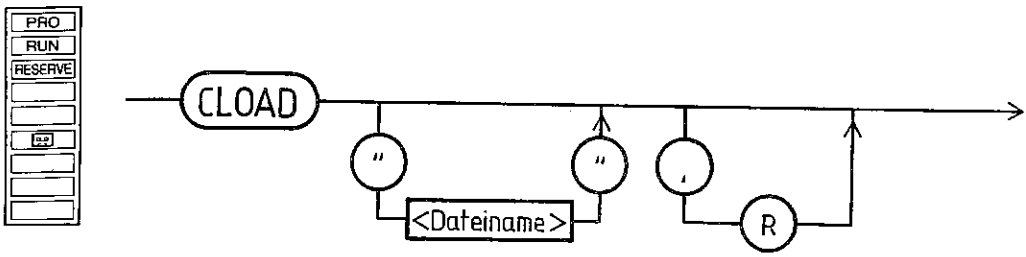
Man kann das CLEAR-Kommando auch innerhalb eines Programmes verwenden. In jedem Falle läßt sich mit ihm Speicherplatz wiedergewinnen, der durch die Erzeugung und Belegung von Variablen für das eigentliche Programm verlorengegangen ist. Zum Beispiel mögen im ersten Teil eines Programmes soviele Variablen verwendet worden sein, daß kein freier Speicherplatz mehr übrig bleibt. Braucht man diese Variablen im zweiten Programmteil nicht mehr, so kann mit CLEAR wieder Platz für neue Variablen geschaffen werden.

BEISPIEL :

```
5:WAIT 30                 'Setzt Wartezeit für PRINT
10:DIM C(5)               'Dimensioniert Array C(N)
20:FOR N= 1 TO 5          'Diese Zeilen lesen die
30:READ A:LET C(N)=A      '  DATA-Werte ein und
40:PRINT C(N)             '  zeigen sie an.
40:NEXT N
50:DATA 10,20,30,40,50'Stellt die Daten bereit
60:CLEAR                  'Löscht alle Variablen
70:PRINT A                'Beweise Löschung
80:END
```

Prüfen Sie nach Ablauf des Programmes, ob auch das Array gelöscht worden ist, indem Sie eine Zuweisung versuchen, z.B: C(2)=99. Existiert das Array nicht, erscheint die Meldung: ERROR 6.

CLOAD



Siehe auch : CLOAD?, CSAVE, MERGE

WIRKUNG : CLOAD lädt BASIC-Programme oder Belegungen für die Funktionstasten von Cassette, die mit CSAVE gesichert worden sind.

HINWEISE : Vor dem Ladevorgang löscht CLOAD grundsätzlich das im Speicher befindliche Programm.

Ohne Parameterangabe wird die nächste auf der Cassette auffindbare Datei geladen. Mit Angabe des Parameters **<Dateiname>** wird zuvor nach der betreffenden Datei auf der Cassette gesucht.

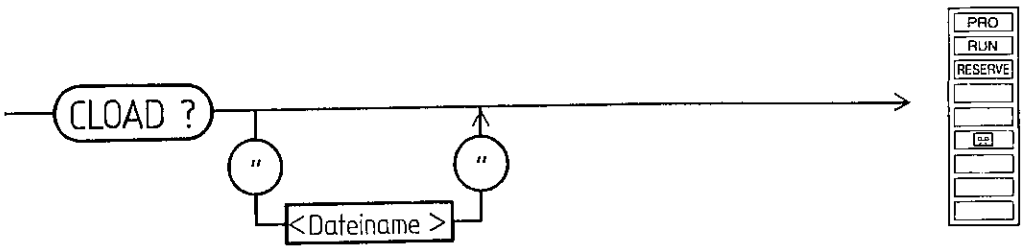
Programme können sowohl im PRO- als auch im RUN-Modus geladen werden. Tastaturbelegungen sind im RESERVE-Modus zu laden. Es ist darauf zu achten, daß der Modus zu der zu ladenden Datei paßt. Ein im RESERVE-Modus geladenes Programm zerstört die derzeitige Tastaturbelegung. Umgekehrt zerstört eine im RUN- oder PRO-Modus geladene Tastaturbelegung das gerade im Speicher stehende Programm.

Ist der CLOAD-Befehl mit der Option R versehen, wird ein geladenes BASIC-Programm automatisch gestartet. Liegt kein BASIC-Programm dabei vor, wird ein ERROR-Code angezeigt.

Eine Fehlermeldung erscheint auch dann, wenn das Programm mit einem Kennwort durch Anwendung des PASS-Befehles geschützt sein sollte.

BEISPIELE : CLOAD
lädt das nächste auffindbare Programm.

CLOAD "P1"
sucht auf der Cassette das Programm P1 und lädt es bei Vorhandensein.



Siehe auch : CLOAD, CSAVE, MERGE

WIRKUNG : Dient dem Vergleich zwischen einer im Speicher und einer auf Cassette befindlichen Datei auf deren Übereinstimmung.

HINWEIS : Diese Verifikation kann sich dabei auf ein im PRO- oder RUN-Modus benutztes Programm beziehen oder auf eine im RESERVE-Modus genutzte Belegung der Funktionstasten.

Wird CLOAD? ohne Parameterangabe angewendet, muß der Anwender für die richtige Positionierung des Cassetten-Bandes sorgen, weil der Vergleich mit der unmittelbar als nächstes auf der Cassette auffindbaren Datei stattfindet.

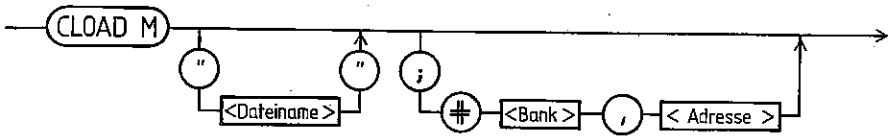
Mit Angabe des Parameters **<Dateiname>** stellt man sicher, daß zum Vergleich nur die genannte Datei von der Cassette herangezogen wird.

Der Vergleich erfolgt blockweise. Sofern sich eine Abweichung zwischen zwei Blöcken ergibt, wird der Ablauf gestoppt und ein ERROR-Code angezeigt. Fällt der Vergleich positiv aus, d.h. stimmen beide Dateien überein, erscheint auf dem Display wieder das Bereitschaftszeichen >.

BEISPIEL :

```
>
CLOAD "PROG01"          Programm laden
                        Cassette zurückspulen
>
CLOAD?"PROG01"          Programm überprüfen
```

CLOAD M



Siehe auch : BLOAD, CALL, CSAVE M, NEW

WIRKUNG : Lädt Maschinensprache-Programme von der Cassette in den Arbeitsspeicher.

HINWEISE : Maschinensprache-Programme unterscheiden sich zu BASIC-Programmen in ihrem Aufzeichnungsformat. Deshalb können sie nicht mit dem Befehl CLOAD geladen werden. Auch werden sie in einem anderen Speicherbereich abgelegt als BASIC-Programme.

<Bank> bestimmt die Speicherbank (0...7), in die das Maschinensprache-Programm zu laden ist.

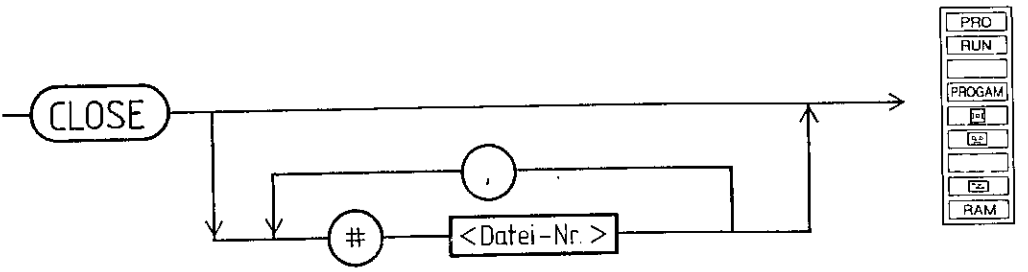
<Adresse> bestimmt dabei, ab welcher Adresse innerhalb der gültigen Speicherbank das Programm abgelegt werden soll.

Diese beiden Parameter können auch weggelassen werden. Wird jedoch einer von beiden angegeben, muß auch der jeweils andere spezifiziert sein. Fehlt dieses Parameterpaar, so wird das Programm in jene Speicherbank ab jener Adresse geladen, wie es sich vor seiner Aufzeichnung im Speicher befunden hat.

Wurde das Programm bei seiner Aufzeichnung mit einer Autostart-Adresse versehen, wird es an eben dieser Adresse nach Abschluß des Ladevorganges automatisch gestartet.

BEISPIEL : CLOAD M "MAC1"

Diese Anweisung lädt von der Cassette das mit MAC1 bezeichnete Maschinensprache-Programm an die Adresse und in die Speicherbank, wo es sich bei der Aufzeichnung zuvor befunden hat.



Siehe auch : END, OPEN

WIRKUNG : CLOSE schließt alle spezifizierten Dateien.

HINWEISE : Mit CLOSE wird die Möglichkeit des Zugriffs auf Dateien beendet, d.h. diese geschlossen.

Ohne Parameterangabe schließt CLOSE alle offenen Dateien. Mit Angabe der Parameter <Datei-Nr.> werden nur die Daten geschlossen, die unter der jeweils gleichen Nummer zuvor mit OPEN zu einem bestimmten Zweck geöffnet worden sind.

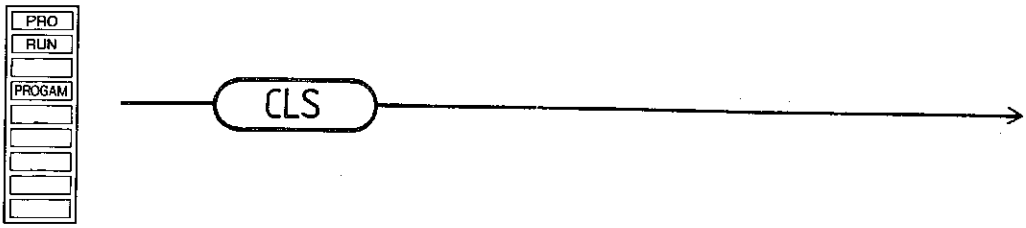
Eine einmal geöffnete Datei muß, bevor sie für einen anderen Zugriffszweck (Eingabe, Ausgabe, Datenanhang) geöffnet werden kann, zuvor mit CLOSE geschlossen werden.

Versucht man, eine bereits geöffnete Datei zu öffnen, wird ein ERROR-Code ausgegeben.

Alle geöffneten Dateien werden automatisch bei Ausführung der Befehle END, NEW, RUN und LOAD und bei Ausschaltung des Computers geschlossen. Eine Dateischließung erfolgt auch dann, wenn man ein Programm editiert.

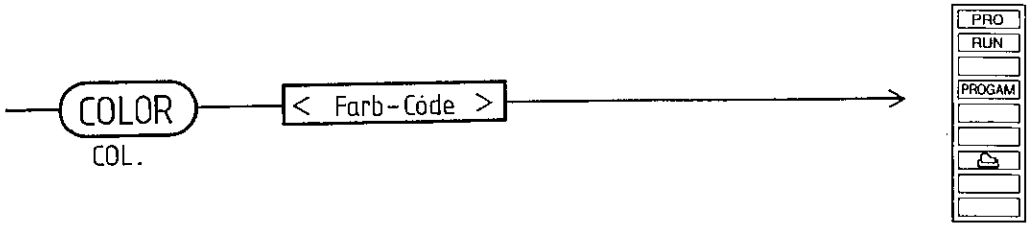
BEISPIEL :

```
5:MAXFILES=2
10:OPEN "X:PAYMENT" FOR INPUT AS #1
20:OPEN "CAS:UPDATE" FOR INPUT AS #2
:
:
400:CLOSE #1,#2
```



WIRKUNG : CLS löscht das Display.

HINWEISE : Das Kommando CLS löscht den Inhalt sämtlicher Display-Zeilen und setzt den Cursor an die linke obere Display-Ecke. Diese "Home-Position" trägt den im MODE 0 Koordinatenpunkt (0,0).



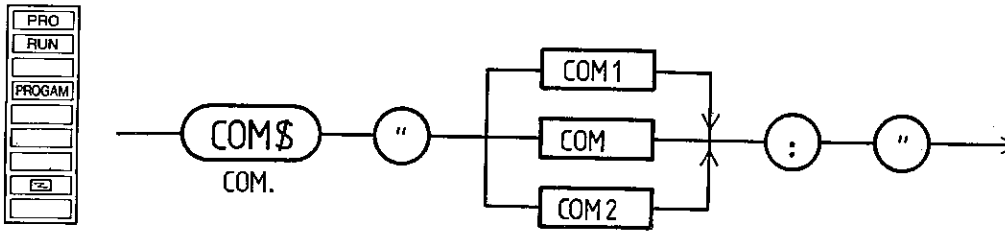
WIRKUNG : COLOR selektiert den gewünschten Farbstift des Druckers CE-1600P (bzw. CE-150).

HINWEISE : Der Stift wird über den Parameter <Farb-Code> bestimmt, wobei folgende Farben möglich sind:

<Farb-Code>	Farbe
0	schwarz
1	blau
2	grün
3	rot

Mit Einschaltung des Computers wird automatisch der schwarze Stift ausgewählt.

COM$



Siehe auch : SETCOM

WIRKUNG : COM$ liefert einen String, der die zuletzt über SETCOM vereinbarten Kommunikations-Parameter des spezifizierten Ports enthält.

HINWEISE : Die in diesem String enthaltenen Parameter sind in genau der Reihenfolge angeordnet, wie sie mit SETCOM festzulegen sind. Sie lautet:

, <WL>, <PR>, <ST>, <XO>, <SI>

COM$ "COM1:" liefert die Einstellungen der RS-232C-Schnittstelle,

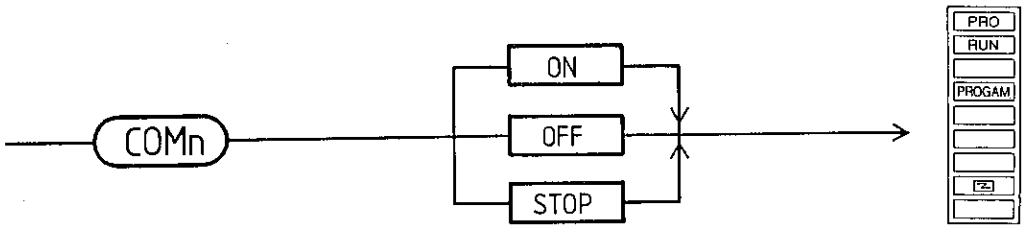
COM$ "COM2:" dagegen die des SIO-Interface (optoelektronischer Port).

COM$ "COM:" meint dasjenige Interface, das zuletzt über SETDEV selektiert worden ist.

BEISPIEL :

```
10:SETCOM "COM1:",300,,,2
20:PRINT COM$ "COM1:"

RUN
300,8,N,2,X,S
>
```



Siehe auch : ON COMn GOSUB, SETCOM

WIRKUNG : Diese Anweisungen erlauben oder verhindern die Annahme von Interrupt-Anforderungen, die über eines der beiden Kommunikations-Ports an den Computer gestellt werden.

HINWEISE : COMn meint, stellvertretend für COM1 und COM2 geschrieben, jeweils folgendes Interface:

COM1 = RS-232C-Interface
COM2 = SIO-Interface

COMn ON erlaubt die Annahme eines ankommenden Interrupts, der über die Anweisung ON COMn GOSUB mit der entsprechend dafür vorgesehen Unterroutine bearbeitet werden kann.

COMn OFF verbietet die Annahme des Interrupts. Die Anweisung ON COMn GOSUB wird dann nicht beachtet.

COMn STOP verbietet ebenfalls die Annahme dieser Interrupts, merkt sich den jeweils zuletzt angeforderten in einem Zwischenspeicher. Bei der nächsten Ausführung von COMn ON wird dieser Interrupt dann unverzüglich abgearbeitet. COMn STOP ist die vom System standardmäßig angenommene Einstellung.

Siehe auch : RESUME, RUN, STOP, WAIT

WIRKUNG : Das Kommando CONT setzt abgebrochene oder unterbrochene Programmabläufe fort.

HINWEISE : Eine solche Fortsetzung ist nur bei folgenden Abbruch- oder Unterbrechungsursachen möglich:

- Abbruch durch STOP-Anweisung
- Abbruch durch Betätigung der Taste BREAK
- Unterbrechung durch PRINT-Anweisung

In diesen Fällen wird CONT jedoch ignoriert:

- Programm abgearbeitet oder durch END beendet
- Programm im PRO-Modus geändert
- Programm mit ERROR-Code abgebrochen

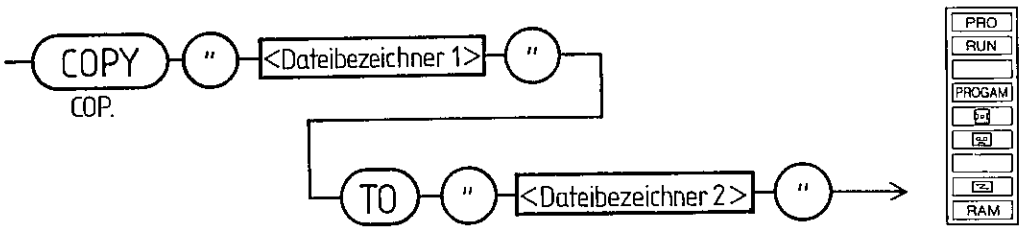
Anstelle des CONT-Kommandos läßt sich auch eine GOTO-Anweisung mit spezifizierter Zeilennummer verwenden. Noch einfacher geht es aber, wenn man schlicht und ergreifend die Taste [↓] betätigt.

BEISPIEL :

```
10:PRINT "PROGRAMM STOPPT HIER"
20:STOP
30:PRINT "PROGRAMM FORTGESETZT"
40:PRINT "PROGRAMM BEENDET
50:END

RUN
PROGRAMM STOPPT HIER
 BREAK IN 20
>

CONT
PROGRAMM FORTGESETZT
PROGRAMM BEENDET
>
```



Siehe auch : SET

WIRKUNG : COPY kopiert eine Datei.

HINWEISE : Es sind drei verschiedene Kopiervorgänge möglich

a) Kopie von einem zum anderen Medium unter Beibehaltung der Dateibezeichnung.

b) Kopie von einem zum anderen Medium unter geänderter Dateibezeichnung.

c) Kopie auf demselben Medium unter einer anderen Dateibezeichnung.

Die in Anführungszeichen zu setzenden Parameter **<Dateibezeichner 1>** und **<Dateibezeichner 2>** sind in folgende Angaben zu unterteilen:

<Dateibezeichner> = <Dn:FILENAMEn.EXTn>

Hierbei soll nach der Ersetzung von n gelten:

D1: Datenquelle (Quellmedium)
D2: Datenziel (Zielmedium)

FILENAME1 Name der Quelldatei
FILENAME2 Name der Zieldatei

EXT1 Extension der Quelldatei
EXT2 Extension der Zieldatei

Bei Anwendung von COPY muß jeder Dateibezeichner mit einer Extension versehen sein. Diese Angabe darf keinesfalls fehlen.

Die Mediumangabe D2 des zweiten Dateibezeichners ist optional und kann also weggelassen werden. Fehlt sie, sind Ziel- und Quellmedium identisch. Es kann dann nur eine Kopie gemäß Fall c) vorgenommen werden, bei dem sich die Namen der Quell- und Zieldatei voneinander unterscheiden müssen.

In einer COPY-Anweisung sind keine "wildcards" zulässig. Bezeichnungen können also nicht durch einen Stern (*) oder ein Fragezeichen (?) als mehrdeutige Benennungen ausgelegt werden.

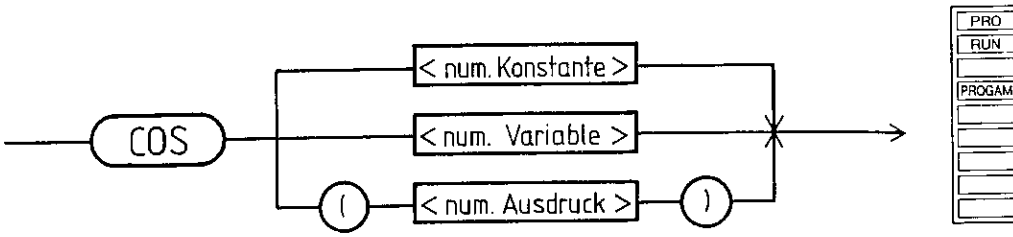
Eine Datei kann nicht als eine Zieldatei kopiert werden, wenn bereits eine Datei gleichen Namens existiert.

Mit COPY sind keine Dateien von einem seriellen Interface auf Diskette, RAM-Disk oder Cassette kopierbar.

Der Befehl COPY kann auch dazu benutzt werden, um eine Datei von einer Diskette auf eine andere Diskette zu kopieren. Als Mediumbezeichnung sind dann die logischen Namen X: und Y: zu verwenden.

BEISPIEL : COPY "S1:RICH" TO "S2:RICH"

Diese Anweisung kopiert die Datei RICH, die sich im Modul des Modulfaches S1 befindet, auf jenes RAM-Modul in Modulfach S2.



Siehe auch : ACS, SIN, TAN

WIRKUNG : Die Funktion COS liefert den Cosinus-Wert eines angegebenen Winkelargumentes.

HINWEISE : Der durch einen numerischen Ausdruck vertretene Winkel kann entweder in Altgrad, Bogenmaß oder Neugrad vorliegen. Damit der Computer den dazugehörigen Funktionswert liefern kann, muß er im passenden Winkelmodus betrieben werden. Hierbei gilt:

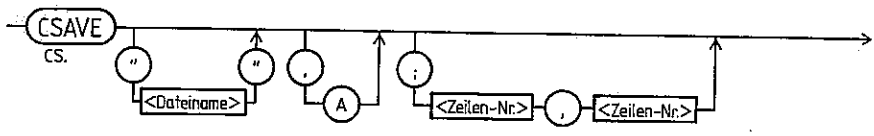
Winkelmaß	Winkelmodus
Altgrad	DEGREE
Bogenmaß	RADIAN
Neugrad	GRAD

BEISPIEL :

```
10:DEGREE
20:G$=CHR$ (&F8)
30:PRINT "cos(60";G$;") = ";COS(60)
40:PRINT "cos(90";G$;") = ";COS(90)
50:END

>RUN
cos(60°) = 0.5
cos(90°) = 0
>
```

CSAVE



Siehe auch : CLOAD, CLOAD?, LLIST, MERGE

WIRKUNG : CSAVE sichert eine Tastaturbelegung, ein BASIC-Programm oder einen Teil davon auf Cassette.

HINWEISE : Bei der Sicherung eines Programmes oder eines Programmteiles muß der RUN- oder der PRO-Modus eingestellt sein. Zur Sicherung einer Tastaturbelegung ist der RESERVE-Modus erforderlich.

CSAVE ohne Parameterangabe sichert die Daten ohne Mitaufzeichnung einer Benennung.

CSAVE "<Dateiname>" sichert die Daten dagegen unter Mitaufzeichnung des angegebenen Namens.

Bei Verwendung der Option A erfolgt die Datenaufzeichnung im ASCII-Format, im anderen Falle im komprimierteren Binär-Format.

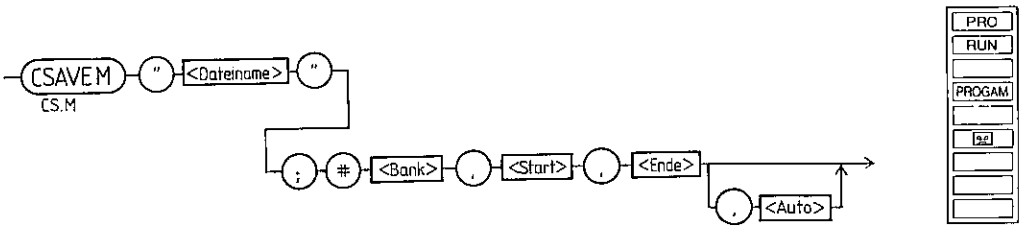
Die erste <Zeilen-Nr.> nennt die Zeilennummer, mit der die Programmaufzeichnung beginnen soll. Die zweite <Zeilen-Nr.> legt die Programmzeile fest, die als letzte mit aufzuzeichnen ist.

Ist das aufzuzeichnende Programm über PASS mit einem Kennwort (password) geschützt, kann keine Aufzeichnung mittels CSAVE erfolgen bevor dieses Kennwort nicht gelöscht worden ist.

Mit CSAVE gesicherte Dateien können nur mit dem Befehl CLOAD geladen werden.

BEISPIEL : CSAVE "PROG1";200,330

Diese Anweisung zeichnet von dem im Speicher befindlichen BASIC-Programm die Zeilen 200 bis einschließlich 330 unter der Bezeichnung PROG1 auf Cassette im binären Format auf.



Siehe auch : CLOAD M, CALL, BSAVE

WIRKUNG : CSAVE M sichert Maschinensprache-Programme auf Cassette.

HINWEISE : **<Dateiname>**

bestimmt, unter welchem Namen das Programm auf Cassette aufgezeichnet werden soll und unter welchem Namen man es dort wieder auffinden und in den Speicher zurückladen kann.

<Bank>

selektiert dabei jene Speicherbank von 0 bis 7, auf der das Programm abgelegt ist.

<Start>

bestimmt jene Adresse, ab der mit der Sicherung des Programmes begonnen und dieses zu höheren Adressen hin fortgesetzt werden soll, bis die Endadresse des Programmes erreicht ist.

<Ende>

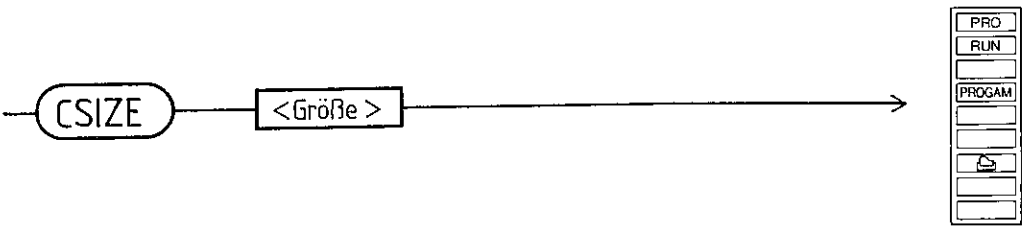
spezifiziert die Adresse, in der das letzte Byte des Maschinensprache-Programmes vorzufinden ist, d.h. bis zu einschließlich welchem Byte die Aufzeichnung des Programmes vorzunehmen ist.

<Auto>

ist optional und bestimmt, mit welcher Adresse das Maschinensprache-Programm nach seiner Rückladung mit CLOAD M automatisch zu starten ist. Fehlt diese Angabe, wird sie standardmäßig zu &FFFF angenommen, was einer Ausschaltung der Autostart-Funktion gleichkommt.

BEISPIEL : CSAVE M "MAC1";#4,&8000,&8AFF

.....zeichnet ein Maschinensprache-Programm, das sich in der Speicherbank 4 befindet und sich im Adreßbereich &8000 bis einschließlich &8AFF erstreckt, auf einer Cassette unter dem Namen MAC1 auf. Wird es zu einem späteren Zeitpunkt in den Speicher zurückgeholt, legt es der Computer an genau der ursprünglichen Stelle ab (sofern in der Anweisung CLOAD M nichts anderes vereinbart wird). Da keine Autostart-Adresse <Auto> bei der Aufzeichnung angegeben worden ist, findet kein automatischer Start des Programmes statt.



Siehe auch : PCONSOLE

WIRKUNG : CSIZE bestimmt, in welcher Größe die Zeichen mit dem Drucker CE-1600P bzw. CE-150 abbildbar sind.

HINWEISE : Die **<Größe>** wird durch einen ganzzahligen Wert im Bereich von 1..9 wie folgt festgelegt:

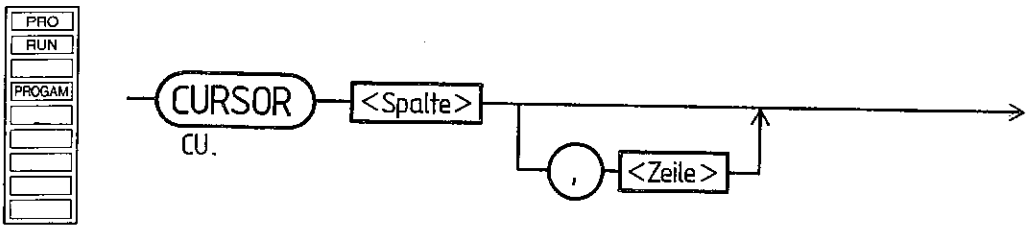
<Größe>	Zeichen/Zeile	Höhe (mm)	Breite (mm)
1	160	1.2	0.8
2	80	2.4	1.6
3	53	3.6	2.4
4	40	4.8	3.2
5	32	6.0	4.0
6	26	7.2	4.8
7	22	8.4	5.6
8	20	9.6	6.4
9	17	10.8	7.2

Die hier gezeigten Zeichengrößen gelten bei dem Wert <Zeilenlänge>=0, mit dem PCONSOLE die Länge der Zeilen begrenzungslos schaltet.

BEISPIEL :

```
10:FOR I=1TO 4
20:CSIZE I
30:LPRINT "SHARP PC-1600"
40:LPRINT
50:NEXTI
60:CSIZE 2
70:END
```

CURSOR



Siehe auch : PRINT

WIRKUNG : CURSOR setzt den Cursor des Displays an die gewünschte Zeichenposition.

<Spalte> bestimmt dabei die gewünschte Spalte (X-Position) im Bereich von 0 bis 25.

```
<Spalte>=0  :  linker Displayrand
<Spalte>=1  :  2.  Spalte von links
<Spalte>=2  :  3.  Spalte von links
    .                  .
    .                  .
<Spalte>=25 : rechter Displayrand
```

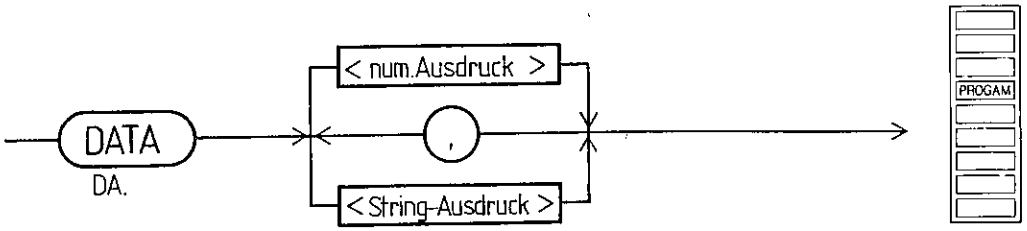
<Zeile> bestimmt dagegen die gewünschte Zeile (Y-Position) im Bereich von 0 bis 3.

```
<Zeile>=0   :  oberste  Zeile
<Zeile>=1   :  zweite   Zeile
<Zeile>=2   :  dritte   Zeile
<Zeile>=3   :  unterste Zeile
```

Diese Parameter wird nur im MODE 0 vom PC-1600 angenommen.

BEISPIEL :

```
10:WAIT 50
20:FOR N=1 TO 6
30:READ A$
40:CURSOR 12,1:PRINT A$
50:CURSOR 12,1:PRINT " "
60:NEXT N
70:WAIT 0
80:END
90:DATA "H","A","L","L","O","!"
```



Siehe auch : READ, RESTORE

WIRKUNG : DATA dient zur Auflistung von numerischen oder String-Konstanten, die mit der READ-Anweisung gelesen werden können.

HINWEISE : Eine DATA-Anweisung kann gleichzeitig numerische und String-Konstante in gemischter Aufzählung enthalten. Die Konstanten müssen jeweils durch ein Komma voneinander getrennt sein.

Mit jeder neuen READ-Anweisung läßt sich immer eine Konstante nach der anderen aus dieser Liste ablesen. Damit das Zusammenspiel zwischen READ und DATA funktioniert, muß die gerade zu lesende Konstante vom selben Typ sein wie die in der READ-Anweisung angegebene Variable.

Welches Element der Liste gerade lesbar ist, wird durch einen internen Zeiger bestimmt, der automatisch nach jedem Lesevorgang entsprechend um eine Position weiter gesetzt wird.

Sind alle in der DATA-Liste enthaltenen Elemente gelesen, kann keine weitere READ-Anweisung ausgeführt werden, bevor nicht eine Rücksetzung des internen Zeigers mit Hilfe des RESTORE-Befehles erfolgt.

DATA gehört wie REM zu den sogenannten nichtausführbaren Anweisungen, was bedeuten soll, daß bei einem Sprung auf eine solche Anweisung der Computer nach der nächsten Anweisung sucht, die nicht mit dem Befehlswort DATA beginnt und dort den weiteren Programmablauf fortsetzt.

DATA-Anweisungen können an beliebiger Stelle im BASIC-Programm vorkommen. Sie brauchen nicht so plaziert zu werden, daß sie vor dem READ-Befehl, der ihre Inhalte liest, stehen. Der Computer sucht sich nämlich die nächste auffindbare und noch nicht abgelesene DATA-Anweisung selbst.

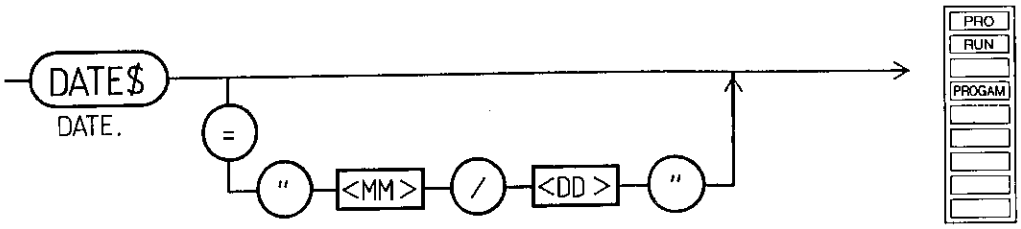
BEISPIELE :

```
10:DATA "MICHAEL",23
20:FOR J=1 TO 5
30:READ A$,B
40:PAUSE A$,B
50:NEXT J
60:END
70:DATA "SUSANNE",-24,"NICOLE",38,"   PETER",57

>
RUN
MICHAEL                 23
SUSANNE                -24
NICOLE                  38
   PETER                57
ERROR 4 IN 30
```

Dieses Programmbeispiel wird mit Ausgabe eines ERROR-Codes beendet, da nach 4 Durchläufen der durch die Zeilen 20 bis 50 gebildeten Programmschleife die nächste READ-Anweisung keine lesbaren Daten mehr vorfindet. Die Programmschleife möchte zwar gerne fünf Datenpaare A$,B lesen, es stehen in den DATA-Anweisungen des Programmes jedoch nur vier solche Datenpaare bereit.

```
10:FOR I=1TO 5
20:READ N
30:PAUSE N
40:NEXT I
50:END
60:DATA 10,2*I,I+N,4,ACS(I/10)
```



Siehe auch : TIME$, ALARM$

WIRKUNG : DATE$ ist eine Systemvariable, die das Datum der eingebauten Echtzeit-Uhr enthält.

HINWEISE : Sie kann dazu verwendet werden, das Datum zu lesen oder aber auch zu setzen.

Stellen des Datums

Um das Datum zu setzen, ist dieser Variablen ein String folgendes Formates zuzuweisen: "MM/DD"

MM bedeutet dabei die zweistellige Monatsangabe im Bereich 01..12.

DD bedeutet dabei die zweistellige Tagesangabe im Bereich 01..31.

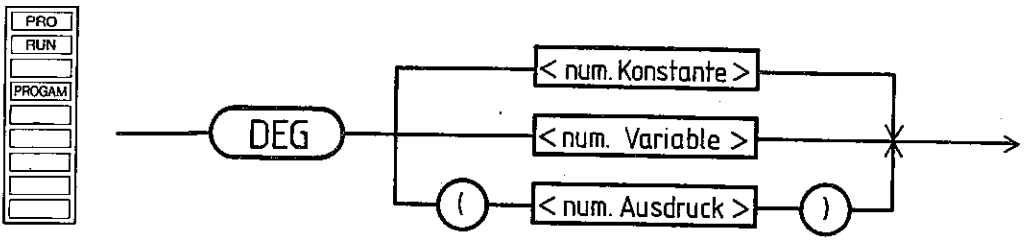
Abfrage des Datums

Zur Abfrage des aktuellen Datums ist DATE$ als Variable ohne Wertzuweisung zu verwenden. Sie liefert einen String, dessen Format dem obig gezeigten entspricht.

Der Tag wird um den Wert 1 erhöht, wenn ein Übergang der in der Variablen TIME$ geführten Uhrzeit von 23:59:59 auf 00:00:00 stattfindet.

BEISPIEL : DATE$= "12/25"

Diese Anweisung setzt das Datum der Echtzeit-Uhr auf den 25. Dezember.



Siehe auch : DMS

WIRKUNG : Wandelt einen Winkel, der in Altgrad gemessen wird, von der sexagesimalen Form, also der Darstellung in Stunden, Minuten und Sekunden, in seine dezimale Entsprechung um.

HINWEISE : Der zu wandelnde Winkel muß dabei im Format hh.mmssrr vorliegen, wobei die Ziffern:

hh die Stunden,
mm die Minuten,
ss die Sekunden und
rr den dezimalen Sekundenrest bestimmen.

Folgende Werte sind dabei einzuhalten:

hh : 0 bis ..
mm : 00 bis 59
ss : 00 bis 59
rr : 00 bis 99

Das Ergebnis wird mit bis zu zehn signifikanten Ziffern angezeigt.

BEISPIEL :

```
10:X=DEG 50.3000
20:PRINT X
30:END

>
RUN
                          50.5
>
```

Siehe auch : RADIAN, GRAD

WIRKUNG : Versetzt den Computer in den Winkelmodus DEGREE. In diesem Modus werden alle Winkelangaben als in Altgrad gegeben angesehen und auch in diesem Maß ausgegeben.

HINWEISE : Zur Kennzeichnung dieser Betriebsart wird in der Status-Zeile das Symbol DEG angezeigt.

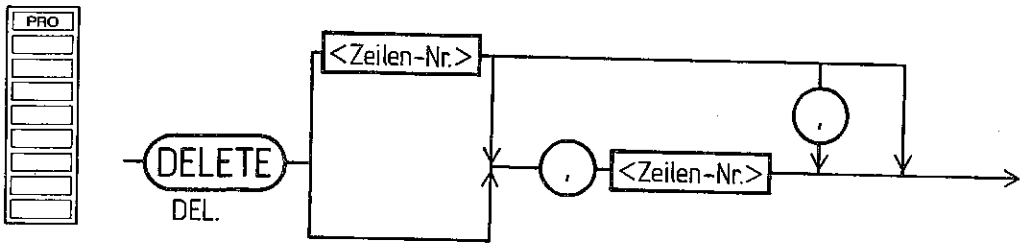
Die Argumente der Funktionen SIN, COS und TAN werden dann als in Altgrad vorliegend angesehen und die Werte der Funktionen ASN, ACS und ATN in dezimalen Altgraden ausgegeben.

BEISPIEL :

```
10:DEGREE
20:PAUSE "WINKELANGABEN IN ALTGRAD"
30:PRINT ASN(0.5),ASN(1)
40:PRINT ACS(0.5),ASN(1)
50:PRINT ATN(0.5),ATN(1)
60:END
```

Lassen Sie dieses Programm zum Vergleich auch in den beiden anderen Winkel-Modi (GRAD und RADIAN) laufen.

DELETE



Siehe auch : NEW

WIRKUNG : DELETE löscht die spezifizierten Zeilen eines BASIC-Programmes.

HINWEISE :

DELETE <Zeilen-Nr.>
löscht genau diese spezifizierte Zeile, sofern sie im Programm vorhanden ist.

DELETE <Zeilen-Nr.>,
löscht, mit der genannten Zeile beginnend, alle weiteren Programm-Zeilen bis zum Programm-Ende.

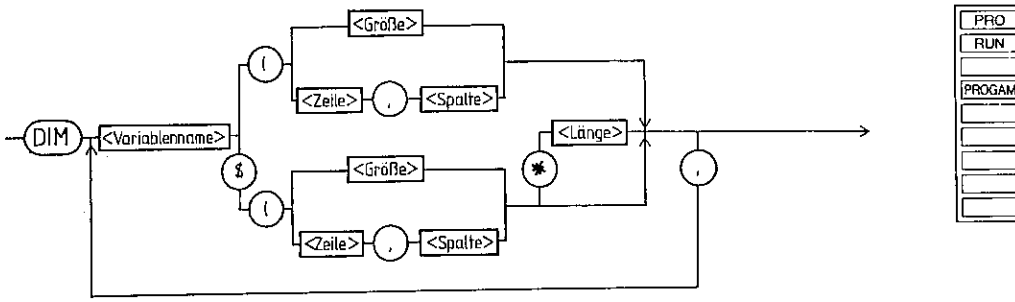
DELETE <Zeilen-Nr.>,<Zeilen-Nr.>
löscht alle Zeilen eines Programmes, beginnend mit der ersten und endend mit der zweiten angegebenen Zeile. Die zweite Zeilennummer muß dabei höherwertiger als die erstgenannte Nummer sein.

DELETE ,<Zeilen-Nr.>
löscht alle Zeilen eines Programmes, beginnend mit der ersten Programmzeile bis einschließlich der spezifizierten Zeile.

Um ein Programm komplett zu löschen, sollte das Kommando NEW verwendet werden.

BEISPIELE :

```
DELETE 150
DELETE 200,
DELETE 50,150
DELETE ,35
```



Siehe auch : CLEAR, ERASE

WIRKUNG : DIM dient der Reservierung von Speicherplatz für die Aufnahme von Array-Variablen des numerischen oder des String-Typs. Daneben bestimmt DIM auch die von einem Array erfaßbare Elemente-Anzahl. Bei String-Arrays läßt sich zusätzlich die Länge der aufzunehmenden Strings definieren.

HINWEISE : Mit Ausnahme der Standard-Variablen A bis Z und A$ bis Z$, die gleichbedeutend mit den beiden eindimensionalen Standard-Arrays @(1) bis @(26) und @$(1) bis @$(26) sind, müssen alle anderen Array-Variablen mit DIM dimensioniert werden, um für diese ausreichend Platz im Speicher bereitzustellen.

Solange diese Dimensionierung fehlt, können die nicht zu den Standard-Variablen zählenden Arrays nicht benutzt werden.

<Größe> bestimmt bei eindimensionalen Arrays die Anzahl der Elemente, die in einem solchen Array, das auch als Liste betrachtet werden kann, aufnehmbar sind. Die zulässigen Werte für <Größe> liegen im Bereich von 0...255, wobei sich die Elemente-Anzahl aus <Größe> + 1 ergibt.

<Zeile> bestimmt bei zweidimensionalen Arrays, die man auch als Tabellen auffassen kann, die Nummer der höchsten vorkommenden Zeile einer solchen Tabelle. Die zulässigen Werte für <Zeile> liegen ebenfalls im Bereich: 0...255.

<Spalte> bestimmt bei zweidimensionalen Arrays die höchste vorkommende Spalte einer solchen Tabelle. Der zulässige Wertebereich ist auch hier: 0...255.

<Länge> bestimmt bei den String-Arrays wie lang die aufzunehmenden Strings sein dürfen. Weisen die Strings jedoch mehr Zeichen auf als mit <Länge> vorgegeben, werden sie auf das entsprechende Maß reduziert und alle überzähligen Zeichen gekappt. Fehlt die Angabe des Parameters <Länge>, können die Strings standardmäßig bis zu 16 Zeichen enthalten. Die maximale Stringlänge beträgt 80 Zeichen.

Die Gesamtheit der Tabellenelemente ergibt sich wegen des Numerierungsbeginnes bei Null aus dem Zusammenhang:

Elementanzahl = (<Zeile> + 1) * (<Spalte> + 1)

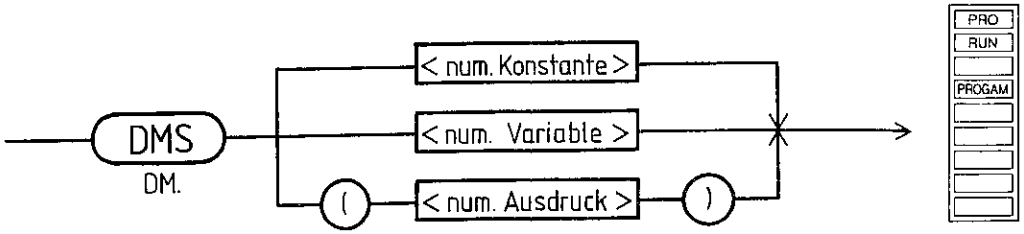
Die Dimensionierung DIM A(2,3) deklariert somit ein Array mit 3 Zeilen und 4 Spalten. Es enthält folgende Tabellenelemente:

Spalte:	1	2	3	4
1. Zeile	A(0,0)	A(0,1)	A(0,2)	A(0,3)
2. Zeile	A(1,0)	A(1,1)	A(1,2)	A(1,3)
3. Zeile	A(2,0)	A(2,1)	A(2,2)	A(2,3)

Nachdem ein Array DIMensioniert worden ist, kann man es nicht umdimensionieren, solange nicht ein Reset des Computers oder aber einer der Befehle CLEAR, NEW, RUN oder ERASE ausgeführt wird. Ein laufendes Programm bricht mit der Ausgabe eines ERROR-Codes ab, wenn es entweder auf ein nicht mit DIM deklariertes Array trifft oder eine DIM-Anweisung vorfindet, die sich auf ein bereits dimensioniertes Array bezieht. Indizes, die die mit <Größe>, <Spalte> oder <Zeile> vereinbarten Maximalwerte überschreiten, führen ebenfalls zu einem Programmabbruch. Negative Indizes sind nicht erlaubt !

BEISPIELE :

```
10:DIM C(13)
20:DIM F$(10)
30:DIM H(4.6)
40:DIM G$(7,5)*25
```



WIRKUNG : DMS wandelt einen in Altgrad vorliegenden Winkel von der dezimalen Darstellungsart in das Format "Stunden,Minuten,Sekunden" um, das man auch als sexagesimale Notation bezeichnet.

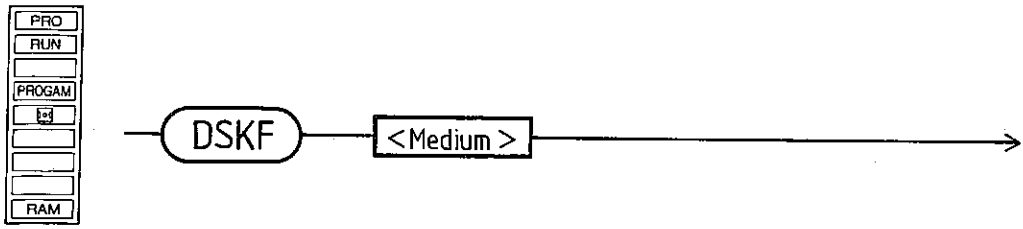
HINWEISE : Das Wandlungsergebnis wird im Format hh.mmssrr ausgegeben, wobei folgendes gilt:

hh	Stunden	00.....
mm	Minuten	00...59
ss	Sekunden	00...59
rr	Dezimaler Sekundenrest	00...99

BEISPIEL :

```
10:X=DMS 50.5
20:PRINT X
30:END

>
RUN
                              -50.3
>
```



WIRKUNG : DSKF liefert den Betrag des noch zur Verfügung stehenden Speicherplatzes auf dem spezifizierten Medium.

HINWEISE : Folgende Spezifikationen können für das <Medium> gewählt werden:

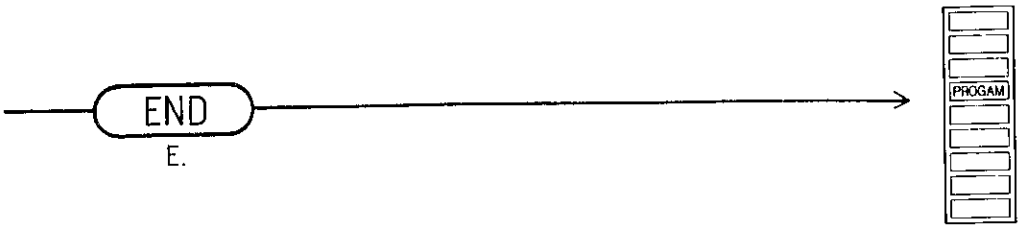
S1:	RAM-Modul im Modulfach S1
S2:	RAM-Modul im Modulfach S2
X:,Y:	Diskette im Diskettenlaufwerk

Der gelieferte Betrag gibt die Anzahl der noch freien Speicherplätze in Bytes an.

BEISPIEL :

```
>
DSKF "S1:"
```

Diese Anweisung liefert die Anzahl der freien Bytes des im Modulfach S1 befindlichen Modules.



Siehe auch : STOP

WIRKUNG : END beendet ein laufendes Programm und schließt alle offenen Dateien und Schnittstellen.

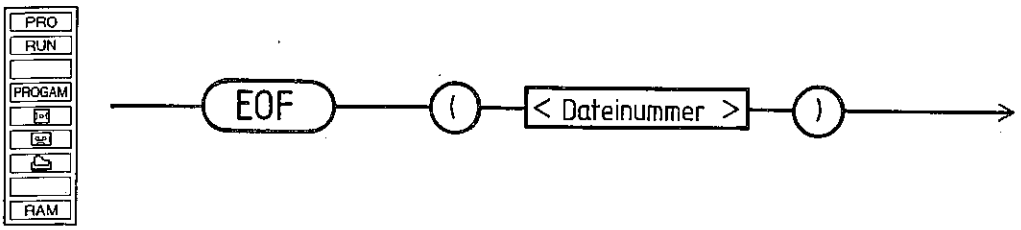
HINWEIS : Es ist nicht zwingend erforderlich, ein Programm mit einer END-Anweisung in der letzten Programmzeile abzuschließen. Bei speziellen Strukturen, die sich bei der Verwendung von Unter-Routinen ergeben, oder in Fällen, wo mehrere voneinander unabhängige Programme in den Speicher zu laden sind, stellt die Einfügung von END-Anweisungen sicher, daß der Computer nicht aus Versehen in unerlaubte Programmteile verzweigt.

Fehlt die END-Anweisung, endet das Programm mit Ausführung der letzten Programmzeile.

BEISPIEL :

```
10:GOSUB 50
20:PRINT "NACH WIEDERKEHR ENDET DAS"
30:PRINT "HAUPTPROGRAMM MIT ZEILE 40"
40:END
50:PRINT "HIER IST DAS UNTERPROGRAMM"
60:RETURN

>
RUN
HIER IST DAS UNTERPROGRAMM
NACH WIEDERKEHR ENDET DAS
HAUPTPROGRAMM MIT ZEILE 40
>
```



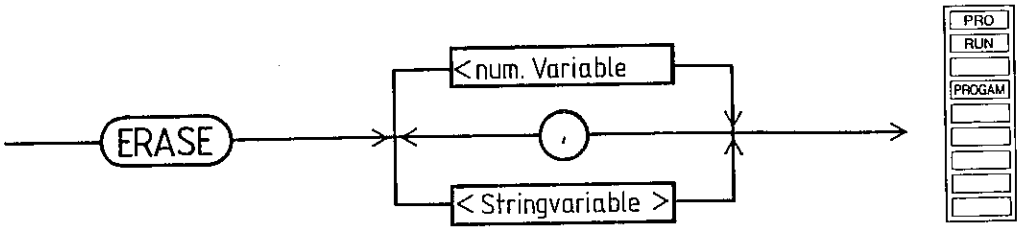
WIRKUNG : EOF liefert einen Wert, aus dem ersichtlich ist, ob beim Lesen einer sequentiellen Datei deren Ende erreicht worden ist.

HINWEISE : Der Parameter **<Dateinummer>** sorgt für die Anwahl der richtigen Datei und muß mit der Nummer übereinstimmen, unter der die Datei geöffnet wurde.

Die möglichen gelieferten Werte sind 0 oder -1. Sie haben folgende Bedeutung:

0	Dateiende noch nicht erreicht
-1	Dateiende erreicht

BEISPIEL : s. Erklärung des OPEN-Befehles.



Siehe auch : CLEAR

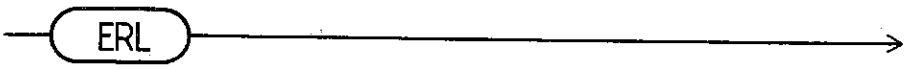
WIRKUNG : ERASE löscht einfache Variablen und Arrays.

HINWEIS : Hierbei sind nur solche numerischen Variablen und String-Variablen löschbar, die nicht zu den Standard-Variablen A bis Z bzw. @(1) bis @(26) und A$ bis Z$ bzw. @$(1) bis @$(26) gehören.

Mit der ERASE-Anweisung lassen sich zwar gezielt einfache Variablen und String-Variablen löschen, jedoch nicht individuelle Elemente eines Arrays. Arrays sind nur komplett löschbar und müssen mit zwei nachfolgenden, aber leeren Klammern kenntlich gemacht sein.

Die zu löschenden Variablen können in Form einer Parameterliste an das Befehlswort ERASE angefügt werden, wobei diese durch Kommas voneinander zu trennen sind.

BEISPIEL : 10:ERASE AB,Z$()

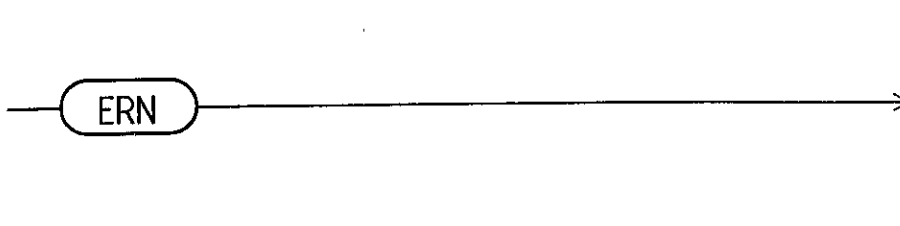


Siehe auch : ERN, ON ERROR GOTO, RESUME

WIRKUNG : ERL liefert die Nummer derjenigen Zeile, in der während des Programmablaufes ein Fehler festgestellt worden ist.

HINWEISE : Die Systemvariable ERL ist dazu gedacht, einen Fehler lokalisieren zu können, wenn durch diesen mittels ON ERROR GOTO Anweisung in eine Fehlerbehandlungs-Routine verzweigt wird. In ERL liegt nur dann eine Zeilennummer vor, wenn der Fehler während eines Programmablaufes aufgetreten ist.

BEISPIEL : s. Erklärung von ERN.



Siehe auch : ERL, ON ERROR GOTO, RESUME

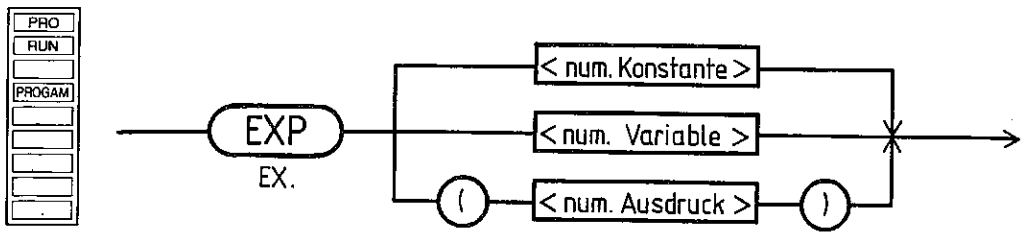
WIRKUNG : ERN liefert den ERROR-Code eines Fehlers, der während eines laufenden Programmes festgestellt worden ist.

HINWEISE : Die Systemvariable ERN wird üblicherweise dazu verwendet, innerhalb einer Fehlerbehandlungs-Routine die Art des aufgetretenen Fehlers feststellen, um geeignete Gegenmaßnahmen ergreifen zu können. Zur genauen Lokalisation des Fehlers wird sie deshalb auch oft gemeinsam mit ERL für die weitere Entscheidungsfindung benutzt.

BEISPIEL :

```
10:ON ERROR GOTO 70
20:FOR N=1TO 20
30:READ A
40:PRINT A
50:NEXT N
60:END
70:PRINT "HABE EINEN FEHLE
   R IN ZEILE";ERL ;" BEME
   RKT:"
80:IF ERL =30AND ERN =4THE
   N PRINT "KEINE DATA-ZEI
   LE GEFUNDEN!"
90:END

>
RUN
HABE EINEN FEHLER IN ZEILE
 30 BEMERKT:
KEINE DATA-ZEILE GEFUNDEN!
```

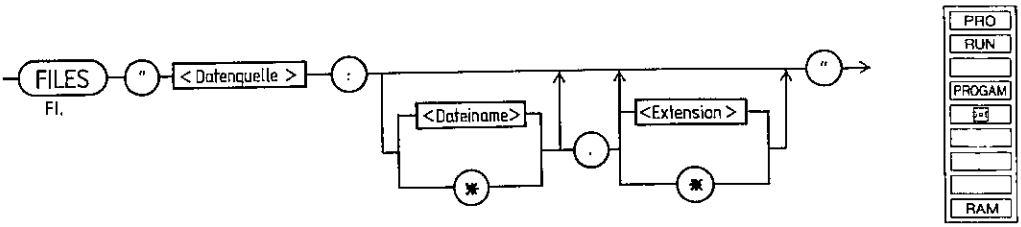


WIRKUNG : Die Funktion EXP(X) liefert zu einem Argument X die zur Basis der Zahl e gebildete Potenz.

HINWEISE : Das Argument X kann eine numerische Konstante oder Variable oder aber ein numerischer Ausdruck sein. Dabei muß X im Wertebereich: -227.9559242 bis +230.2585092 liegen. Werte außerhalb dieses Bereiches liefern den Funktionswert 0.

Die transzendente Basiszahl e wird intern mit 2.7181828 angenähert.

BEISPIEL :
```
PRINT EXP(10)
                220026.46579
>
```



Siehe auch : LFILES, SET

WIRKUNG : FILES liefert eine Liste der auf einer Diskette oder RAM-Disk befindlichen Dateien, also ein Inhaltsverzeichnis dieser Speichermedien.

HINWEISE : Das Inhaltsverzeichnis zeigt jede Datei unter Nennung folgender Einzelinformationen an:

- Dateiname
- Extension (z.B.: .BAS für BASIC-Programme)
- eventuellem Schreibschutzstatus
- Datum der Aufzeichnung
- Uhrzeit der Aufzeichnung

Die gewünschte Liste wird durch den anzugebenden Parameter <Dateibezeichner> bestimmt. Dieser ist in obigem Diagramm aufgeschlüsselt dargestellt. Er besteht aus den Anteilen:

<Datenquelle>, <Dateiname> und <Extension>.

Der Parameter **<Datenquelle>**: meint eines der nachstehenden Speichermedien:

X:	Diskette
Y:	andere Diskette
S1:	RAM-Disk in Modulfach S1
S2:	RAM-Disk in Modulfach S2

Der **<Dateiname>** bestimmt, welche Datei(en) in der Auflistung erscheinen sollen. Dieser Name kann aus bis zu acht Zeichen bestehen. Ersetzt man ihn durch einen Stern, werden alle Dateien, die dieselbe <Extension> aufweisen, angezeigt. Er ist damit beliebig mehrdeutig angegeben. Mit Hilfe von Fragezeichen läßt sich der <Dateiname> auch teilweise mehrdeutig machen.

Da das Display nur eine begrenzte Anzahl Dateien anzeigen kann, lassen sich mit der Taste [↓] die weiteren Dateiangaben Stück für Stück auflisten. Durch Betätigung jeder beliebigen Taste (ausgenommen: [SHIFT], [CTRL], [DEF] und [SML]) läßt sich die Auflistung abbrechen.

Die <Extension> bestimmt, welche Art Datei(en) anzuzeigen sind. So bewirkt beispielsweise die Angabe .BAS die Anzeige von BASIC-Dateien. Die <Extension> kann ebenfalls durch einen Stern oder aber mit Fragezeichen mehrdeutig ausgelegt werden.

Mehrdeutige Dateispezifikationen

Durch Anwendung der Joker-Zeichen (wildcards) * und ? lassen sich wie folgt mehrdeutige Dateibezeichnungen angeben und damit mehrere Dateien ansprechen:

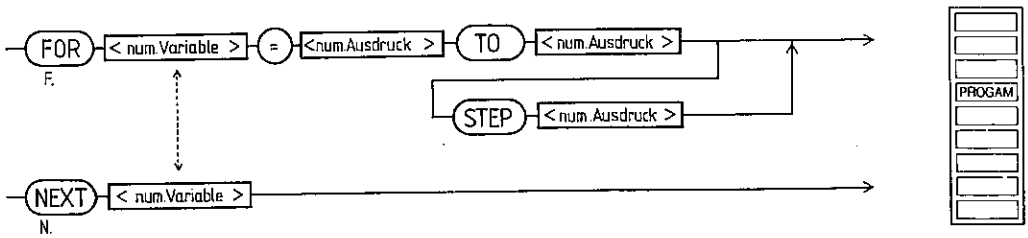
Dateispezifikation	Beispiele passender Dateien			
TEST?	TEST	.BAS	TEST1	.PPP
T??T	TEST	.BAS	TEXT	.BAS
S?MPLE.BIN	SIMPLE	.BIN	SAMPLE	.BIN
A?????	ABCDEF	.	APPLES	.XXX
?A*	RATES	.FIN	RANDOMLY	.GEN
*.XYZ	SHARP	.XYZ	PC1600	.XYZ
NEW.A?C	NEW	.ABC	NEW	.A5C

Eine Mehrdeutigkeit kann auch dadurch erreicht werden, indem man den <Dateinamen> oder aber die <Extension> oder aber beide wegläßt. Allerdings kann bei vorhandener <Extension> der <Dateiname> nicht entfallen, er muß dann durch einen Stern repräsentiert werden.

Grundlegende Parameter-Formate

"<Datenquelle>:"
"<Datenquelle>:<Dateiname>"
"<Datenquelle>:*.<Extension>"
"<Datenquelle>:*.*"
"<Datenquelle>:<Dateiname>.<Extension>"

"<Datenquelle>:<Dateiname>." ist ein besonderes Format und sucht nach Dateien des angegebenen Namens, die keine <Extension> aufweisen.



WIRKUNG : Mit FOR und NEXT lassen sich Programmschleifen bilden und damit Anweisungen mehrfach in vorbestimmter Weise (determiniert) ausführen.

HINWEISE : Der erste Parameter ist eine <num. Variable> und bestimmt, welche Variable als Schleifenzähler (Laufvariable) dienen soll.

Der zweite Parameter weist dieser Laufvariablen einen Anfangswert zu, der durch jeden beliebigen <num. Ausdruck> gebildet sein kann.

Der dritte Parameter gibt den Endwert der Laufvariablen an. Ist er über- oder unterschritten, wird die Programmschleife verlassen und mit der Anweisung nach dem NEXT-Befehl fortgefahren. Dieser Parameter kann ebenfalls ein beliebiger <num. Ausdruck> sein. Er muß im Werte-Bereich -32768...32767 liegen.

Der vierte Parameter ist optional und gibt die Schrittweite, also den Betrag an, der nach jedem Schleifendurchlauf zur Laufvariable hinzugezählt wird (unter Berücksichtigung des Vorzeichens). Die Schrittweite kann nur ganzzahlig sein. Werte mit Nachkommstellen werden auf Integer-Werte gerundet. Der Wert Null ist nicht als Schrittweite erlaubt und hat die Anzeige des ERROR-Codes 19 zur Folge. Wegen der soeben erwähnten Abrundung sind Werte, die dem Betrage nach kleiner als 1 sind, unzulässig, da sie gerundet den Wert Null ergeben. Der zulässige Werte-Bereich ist damit -32768...-1 und 1...32767. Fehlt die Angabe des STEP-Wertes, gilt für ihn standardmäßig 1.

Trifft der Computer auf einen NEXT-Befehl, sucht er sich diejenige zuvor stehende FOR-Anweisung, die mit derselben Laufvariable behaftet ist, wie sie in der NEXT-Anweisung spezifiziert ist. Dann addiert er die Schrittweite zum derzeitigen Wert der Laufvariablen und überprüft, ob ihr Endwert "überschritten" wird. Bei positiver Schrittweite überprüft er, ob der neue Variablenwert größer als der Endwert ist und bei negativen, ob dieser kleiner als der Endwert ist. Trifft dieses zu, wird die Programmschleife verlassen. Im anderen Falle werden die in der Schleife aufgeführten Anweisungen erneut ausgeführt: diesmal mit dem neuen Wert der Laufvariablen.

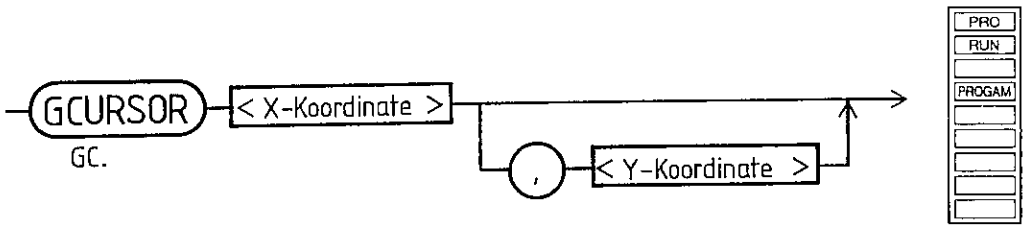
Damit der Computer die richtige FOR-Anweisung finden kann, muß in der NEXT-Anweisung die zugehörige Laufvariable als Parameter angegeben werden.

```
10:FOR I=1 TO 20
:
50:NEXT I
:
:
230:FOR K=2 TO 17 STEP 2
:
290:NEXT K
:
```

FOR-NEXT-Schleifen lassen sich auch ineinander verschachteln:

```
10:FOR M=1 TO 10
20:FOR N=5 TO 20 STEP 5
:
:
80:NEXT N
90:NEXT M
```

FOR-NEXT-Anweisungen treten immer paarweise auf. Zu jeder FOR-Anweisung muß in logischer Folge eine passende NEXT-Anweisung vorzufinden sein. Ist diese Bedingung nicht gegeben, weil ein Teil des Paares fehlt oder aber durch eine verkehrte Schachtelung vorliegt, stoppt der Computer das Programm und gibt einen ERROR-Code aus.



WIRKUNG : GCURSOR positioniert den Grafik-Cursor auf dem gewünschten Display-Punkt.

HINWEISE : Im Grafik-Modus läßt sich das Display als Matrix mit 156 x 32 einzeln adressierbaren Punkten ansprechen. Der Grafik-Cursor kann hierbei nicht nur auf eine innerhalb dieser Matrix liegende Koordinate positioniert werden, sondern auch außerhalb davon. Nach seiner Positionierung kann mit der GPRINT-Anweisung mit der Ausgabe von Grafik-Mustern begonnen werden.

Soll der Cursor im sichtbaren Bereich liegen, so sind folgende Koordinaten einzuhalten:

<X-Koordinate>	:	0...155
<Y-Koordinate>	:	0....31

Beide Koordinaten können jedoch auch Werte im Bereich von -32768 bis 32767 annehmen.

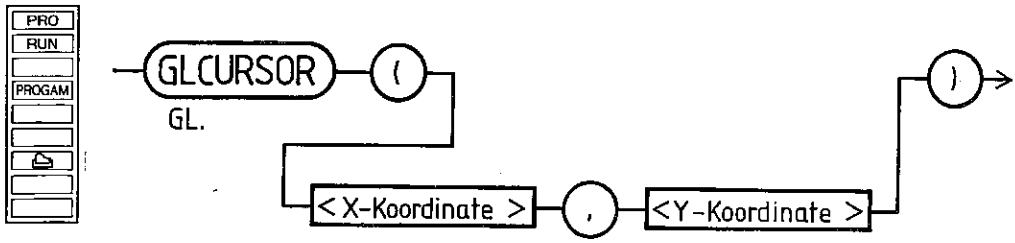
Wird die Angabe der Y-Koordinate weggelassen, gilt die derzeit gültige Y-Position des Grafik-Cursors.

Im Anzeigemodus MODE 1 (PC-1500 Modus) ist der Y-Parameter bedeutungslos und sollte daher in dieser Betriebsart nicht angegeben werden.

BEISPIEL :

```
:
300:X=5
310:Y=20
320:GOSUB 600
:
590:END
600:GCURSOR
610:GPRINT 255,255,255,255
620:RETURN
```

GLCURSOR



Siehe auch : LCURSOR

WIRKUNG : GLCURSOR bewegt im Grafik-Modus den Schreibstift des Druckers an die spezifizierte Position.

HINWEISE : Im Unterschied zum Kommando LCURSOR wird hier der Stift nicht an eine bestimmte Zeichenspalte gesetzt, sondern an die vorgegebene Position, die durch ein Koordinatenpaar (X,Y) definiert ist. Dieser Punkt kann somit auch ein Punkt innerhalb einer Zeichenspalte sein.

Von einem definierten Punkt (X,Y) bestimmmt:

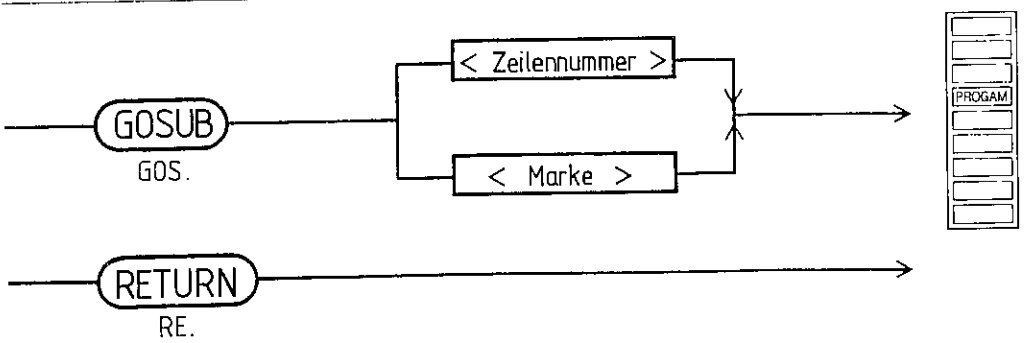
X den horizontalen und
Y den vertikalen Abstand des Punktes zum

derzeit gültigen Koordinatenursprung (Origo).

Die Parameter, <X-Koordinate> und <Y-Koordinate> dürfen wertemäßig im Bereich von -2048 bis 2047 angegeben werden.

Während der Fahrt zur angegebenen Position wird der Stift angehoben.

BEISPIEL : Siehe LLINE



Siehe auch : GOTO, ON..GOSUB, ON..GOTO

WIRKUNG : GOSUB ruft das durch eine <Zeilennummer> oder eine <Marke> spezifizierte Unterprogramm auf.

HINWEISE : Ein Unterprogramm (subroutine) ist eine Gruppe von aufeinanderfolgenden Programmzeilen, die im Ablauf des Gesamtprogrammes mehrfach benötigt werden. Mit Hilfe von Unterprogrammen läßt sich also gewaltig Speicherplatz sparen, wenn man die wiederkehrenden Anweisungen nicht entsprechend oft in das Programm einfügt, sondern sie einmal zusammengefaßt im Speicher ablegt und sie stellvertretend durch GOSUB-Anweisungen aufruft.

Jedes Unterprogramm muß mit dem Befehl RETURN abgeschlossen sein, damit eine Rückkehr in das aufrufende Hauptprogramm möglich ist und der normale Programmablauf hinter der Anweisung GOSUB fortgesetzt werden kann.

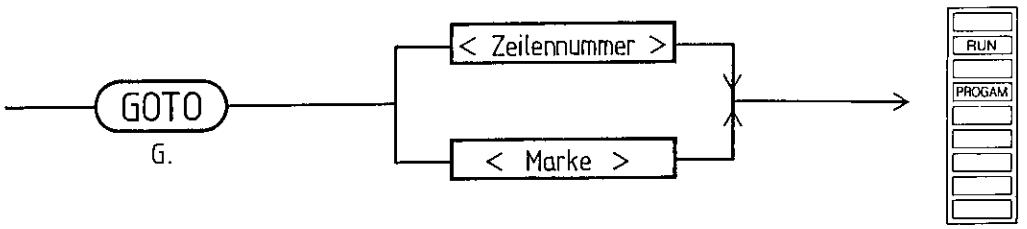
Ein Unterprogramm läßt sich beliebig oft von diversen GOSUB-Anweisungen aufrufen, wobei es selbst wiederum andere Unterprogramme aufrufen kann und so fort. Somit entstehen verschachtelte Unterprogramme, von denen der PC-1600 eine Tiefe von bis zu 15 Ebenen verarbeiten kann. Ist diese Verschachtelungstiefe größer, wird das laufende Programm abgebrochen und ein ERROR-Code auf dem Display angezeigt. Das fehlerhafte Programm muß dann so umstrukturiert werden, daß eine solche Schachtelungstiefe vermieden wird.

Unterprogramme können sich auch selbst aufrufen, wenn man das Programm entsprechend geschickt auslegt und dafür sorgt, daß das Unterprogramm verlassen werden kann und die zulässige Tiefe der Verschachtelung nicht überschritten wird.

BEISPIEL :

```
10:WAIT 100
15:PRINT "HAUPTPROGRAMM GESTARTET"
20:GOSUB 90
25:PRINT "ZURUECK IM HAUPTPROGRAMM"
30:GOSUB 60
35:PRINT "ZURUECK IM HAUPTPROGRAMM"
40:WAIT 0
45:PRINT "PROGRAMM BEENDET"
50:END
55:REM **** UNTERPROGRAMME *****
60:REM --- UNTERPROGRAMM 1 ---
65:PRINT "UNTERPROGRAMM 1 GESTARTET"
70:GOSUB 90
75:PRINT "ZURUECK IM UNTERPROGRAMM 1"
80:RETURN
85:REM --- UNTERPROGRAMM 2 ---
90:PRINT"UNTERPROGRAMM 2 GESTARTET"
95:RETURN

>RUN
HAUPTPROGRAMM GESTARTET
UNTERPROGRAMM 2 GESTARTET
ZURUECK IM HAUPTPROGRAMM
UNTERPROGRAMM 1 GESTARTET
UNTERPROGRAMM 2 GESTARTET
ZURUECK IM UNTERPROGRAMM 1
ZURUECK IM HAUPTPROGRAMM
PROGRAMM BEENDET
>
```



Siehe auch : GOSUB..RETURN, ON..GOTO, CONT

WIRKUNG : GOTO verzweigt den weiteren Programmablauf zu der durch eine <Zeilennummer> oder einer <Marke> spezifizierten Zeile.

HINWEISE : Die GOTO-Anweisung führt einen nichtbedingten Sprung aus, d.h., der Sprung zur spezifizierten Zeile wird unweigerlich ausgeführt und hängt von keiner Bedingung ab (es sei denn man wählt die Anweisung IF..THEN GOTO).
Im Gegensatz zu GOSUB erfolgt kein Rücksprung an die nach GOTO folgende Anweisung.

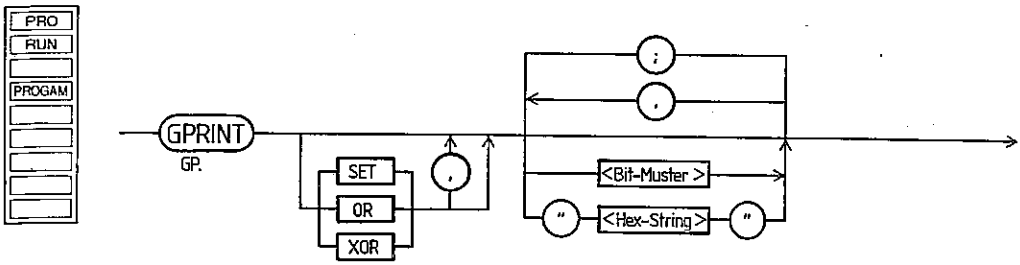
Wird als Sprungziel eine Zeile bestimmt, die die nichtausführbaren Befehle DATA oder REM enthält, wird der Programmablauf mit der nächsten Zeile (bzw. ausführbaren Anweisung) fortgesetzt. Bei nicht vorhandenem Sprungziel erfolgt ein Abbruch des Programmes mit Anzeige eines ERROR-Codes.

Im RUN-Modus läßt sich mit der GOTO-Anweisung auch ein Programm ab einer speziellen Zeile starten. Im Gegensatz zum Kommando RUN werden hierbei keine Variablen gelöscht.

Mit der Anweisung GOTO kann auch ein Programm, welches mit der BREAK-Taste unterbrochen worden ist, fortgesetzt werden. (Siehe auch: CONT).

Eine GOTO-Anweisung ohne Parameterangabe startet das Programm mit der ersten vorhandenen Zeile.

BEISPIEL :
```
10:INPUT A$
20:IF A$="J" THEN 50
30:PRINT "NEIN" : GOTO 50
40:PRINT "JA"
50:END
```



Siehe auch : GCURSOR

WIRKUNG : GPRINT zeichnet Grafik-Muster auf dem Display.

ANMERKUNGEN: Mit GPRINT können jeweils acht direkt übereinanderliegende Display-Punkte durch geeignete Bit-Muster beeinflußt und somit Grafiken auf dem Display abgebildet werden. Jede vertikale Punktreihe hat dabei die Höhe eines im Text-Modus dargestellten Zeichens. Wie sich der Einfluß der Bit-Muster, die je aus 8 Bits bestehen und damit einem Byte entsprechen, auswirkt, hängt von den optionalen Parametern SET, OR und XOR ab.

SET überschreibt die bisher auf dem Display befindliche Anzeige und bildet damit das Bit-Muster originalgetreu ab.

OR verknüpft das definierte Bitmuster mit den momentanen Punktzuständen unter Anwendung der logischen Oder-Funktion (OR function). Alle dunklen Punkte bleiben unverändert. Die hellen Punkte bleiben jedoch nur dann hell, wenn die ihnen zugeordneten Bits des Bitmusters gelöscht sind. Im anderen Falle werden die betreffenden Punkte gesetzt, also dunkelgesteuert.

XOR verknüpft das definierte Bitmuster und das auf dem Display befindliche Punktmuster mit der Exklusiv-Oder-Funktion (XOR function).

Fehlt die Parameterangabe SET, OR oder XOR, so wird standardmäßig die SET-Funktion angenommen.

Die **<Bit-Muster>** können in dezimaler oder aber hexadezimaler Aufzählung angegeben werden, wobei sie jeweils durch ein Semikolon voneinander zu trennen sind. Hexadezimale Werte sind dabei wie üblich durch ein vorangestelltes Kaufmannsund & zu kennzeichnen. Desweiteren lassen sich diese Muster auch mit einem **<Hex-String>** vereinbaren, wobei jedes Byte zwischen 00 und FF stets mit zwei Ziffern ohne Verwendung des & darzustellen ist. Der Verbund der so aufgeführten Bytes ist dann in Anführungsstriche einzuschließen. Wird innerhalb eines solchen Strings eine Ziffer vergessen und damit seine Zeichenanzahl ungerade, so interpretiert der Computer alle Zweiergruppen an Ziffern als gültiges Byte und läßt die letzte verbleibende Ziffer unbeachtet.

Nachstehende Anweisungen sind in ihrer Wirkung identisch:

```
GPRINT 16;40;18;253;18;40;16          (dezimal)
GPRINT &10;&28;&12;&FD;&12;&28;&10 (hexadezimal)
GPRINT "102812FD122810"               (Hex-String)
```

GPRINT ohne jegliche Parameterangabe setzt den Grafik-Cursor um eine Punktzeile nach unten ohne dabei den auf dem Display befindlichen Inhalt zu beeinflussen. Schließt eine GPRINT-Anweisung mit einem Semikolon ab, so bleibt der Grafik-Cursor an der letzten Position stehen. Die nächste GPRINT-Anweisung setzt sich dann an dieser Stelle fort. Wählt man ein Komma als Trennzeichen, so wird zwischen die so getrennten Punktmuster eine Lücke von einer Punktbreite gesetzt.

Beispiel des Zusammenhanges zwischen Punkt- und Bit-Muster, gezeigt für die Option SET:

Punkte	Bit	Zustand		
	0	0		
x	1	1	76543210	<-Bit
x	2	1	¦¦¦¦¦¦¦¦	
	3	0	11010110	Binärwert
x	4	1		
	5	0	&D5	Hex-Wert
x	6	1		
x	7	1	214	Dezimalwert

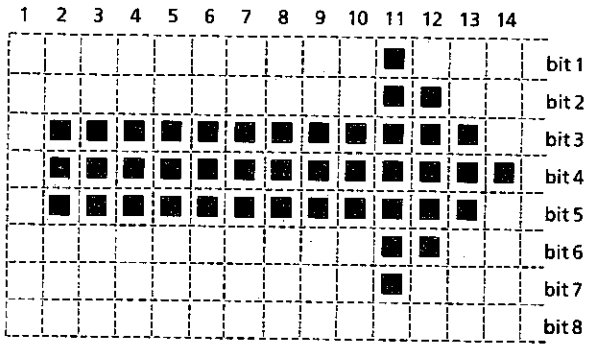


Abbildung A

Das in obiger Abbildung gezeigte Grafik-Symbol kann auf dem Display durch die Ausgabe von 14 Bit-Mustern erzeugt werden. Dabei gilt:

```
Spalte  1: 00000000 = &00
Spalte  2: 00011100 = &1C
         :
         :
Spalte 10: 00011100 = &1C
Spalte 11: 01111111 = &7F
Spalte 12: 00111110 = &3E
Spalte 13: 00011100 = &1C
Spalte 14: 00001000 = &08
```

Mit folgender Anweisung könnte diese Grafik also zur Anzeige gebracht werden:

```
GPRINT &0;&1C;&1C;&1C;&1C;&1C,&1C;
       &1C;&1C;&1C;&7F;&3E;&1C;&8
```

BEISPIEL :

```
10:CLS
20:FOR I=1 TO 77
30:GPRINT 255;214;
40:NEXT I
50:END
```

Siehe auch : DEGREE, RADIAN

WIRKUNG : Versetzt den Computer in den Winkelmodus GRAD. In diesem Modus werden alle Winkelangaben als in Neugrad gegeben angesehen und auch in diesem Maß ausgegeben.

HINWEISE : Zur Kennzeichnung dieser Betriebsart wird in der Status-Zeile das Symbol GRAD angezeigt.

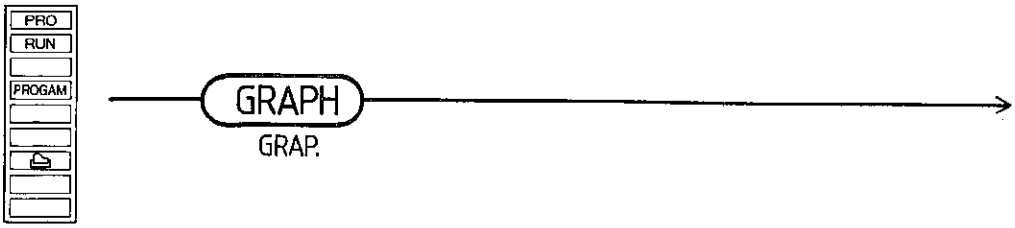
Alle Argumente der Funktionen SIN, COS und TAN werden als in Neugrad vorliegend angesehen und die Werte der Umkehrfunktionen ASN, ACS und ATN im Winkelmaß Neugrad geliefert.

BEISPIEL :

```
10:GRAD
20:PAUSE "WINKEL IN NEUGRAD"
30:PRINT ASN(0.5),ASN(1.0)
40:PRINT ACS(0.5),ACS(1.0)
50:PRINT ATN(0.5),ATN(1.0)
60:END

>
RUN
WINKEL IN NEUGRAD
  33.33333333          100.
  66.66666667            0.
  29.51672353           50.
>
```

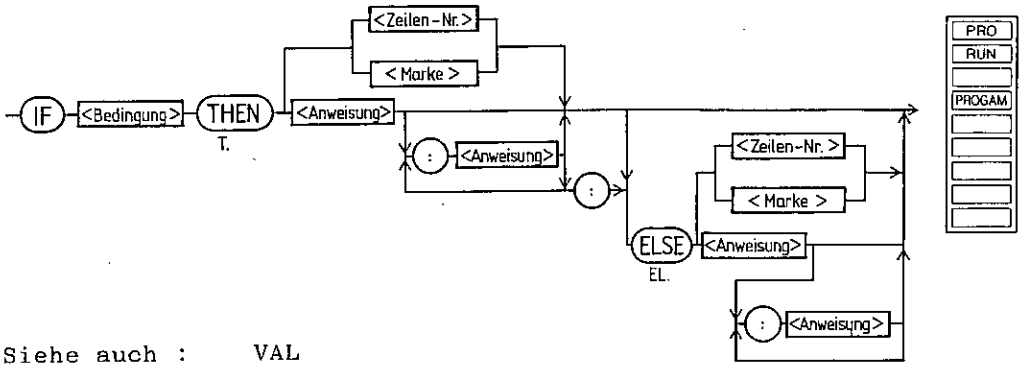
Probieren Sie dieses Programm auch in den beiden anderen Winkel-Modi (DEGREE und RADIAN) aus.



Siehe auch : TEXT

WIRKUNG : Setzt den Drucker (CE-1600P bzw.CE-150) in den Grafik-Modus.

HINWEIS : Mit Ausführung von GRAPH werden die Parameter <Limit 1> und <Limit 2> des Befehles PAPER mit ihren standardmäßigen Werten belegt.



Siehe auch : VAL

WIRKUNG : Die Funktion HEX$ liefert einen String, der sich aus den Hex-Ziffern des als Hexadezimalwert aufgefaßten Argumentes zusammensetzt.

HINWEISE : Das Argument muß ganzzahlig sein und dezimal betrachtet im Bereich 0....65535 liegen. Der gelieferte String ergibt sich damit im Bereich von &0....&FFFF.

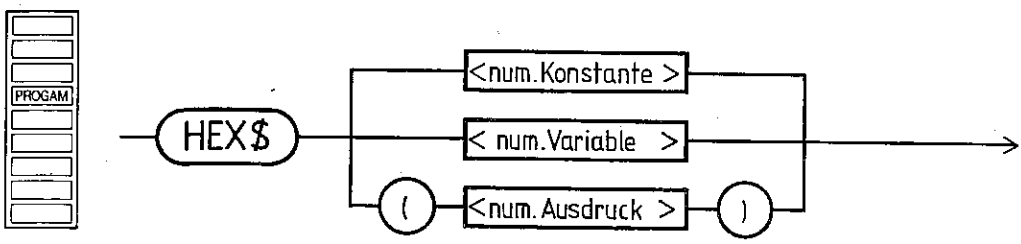
```
HEX$(64)    liefert den String:  "&40"
```

Will man eine hexadezimale Konstante in einen String umwandeln, muß ihr ein sogenanntes Kaufmannsund & vorangestellt werden.

```
HEX$(&E0)   liefert den String:  "&E0"
```

BEISPIEL :

```
10:PRINT "WANDLUNG: DEZIMAL ZU HEX"
20:INPUT "    DEZIMAL-ZAHL = ";X
30:IF X>65535 THEN 100
40:IF X<0 THEN 110
50:PRINT "HEXADEZIMAL-WERT = ";HEX$(X):PRINT
60:INPUT "NOCH EINE ZAHL (J/N) ";A$
70:IF A$="J" THEN 20
80:IF A$="N" THEN END
90:GOTO 60
100:PRINT "FEHLER: MAXIMUM=65535 !":GOTO 20
110:PRINT "FEHLER: MINIMUM=0 !":GOTO 20
120:END
```



WIRKUNG : IF...THEN...ELSE entscheidet über den weiteren Programmverlauf, je nachdem, ob eine Bedingung erfüllt ist oder nicht.

HINWEISE : Die Entscheidung hängt dabei von einer zwischen den Befehlswörtern IF...THEN zu überprüfenden **<Bedingung>** ab. Ist diese erfüllt, wird mit der hinter dem Befehlswort THEN angegebenen Zeile, die durch eine **<Zeilen-Nr.>** oder eine **<Marke>** spezifiziert ist, oder den dortigen stehenden **<Anweisungen>** fortgefahren. Im anderen Falle wird die nächste Zeile ausgeführt.

Enhält die Anweisung IF...THEN den Zusatz ELSE, so wird bei nicht erfüllter Bedingung das Programm nicht mit der nächsten Zeile fortgesetzt, sondern mit der hinter ELSE spezifizierten Zeile oder den dortigen Anweisungen.

Folgt dem Befehlswort ELSE keine Zeilenspezifikation, werden alle Anweisungen (auch die durch einen Doppelpunkt voneinander getrennten) ausgeführt, solange diese sich in derselben Programmzeile befinden.

Fehlt der Zusatz ELSE, so gilt das eben gesagte für das Befehlswort THEN.

Die Anweisungen IF..THEN..ELSE können innerhalb einer Programmzeile von maximal 80 Zeichen Länge auch ineinander verschachtelt werden.

Die zwischen den Befehlswörtern IF...THEN abzufragende **<Bedingung>** wird durch einen **logischen Ausdruck** gebildet, der sich, so komplex er auch immer sein mag, stets aus folgenden Grundformen aufbauen läßt:

X=Y	Vergleich auf Gleichheit
X<>Y	Vergleich auf Ungleichheit
X<Y	Vergleich, ob X kleiner als Y
X>Y	Vergleich, ob X größer als Y
X<=Y	Test, ob X kleiner oder gleich Y.
X>=Y	Test, ob X größer oder gleich Y.
X AND Y	Logische UND-Verknüpfung
X OR Y	Logische ODER-Verknüpfung
NOT X	Logische Verneinung

Beispiele logischer Ausdrücke:

X=1

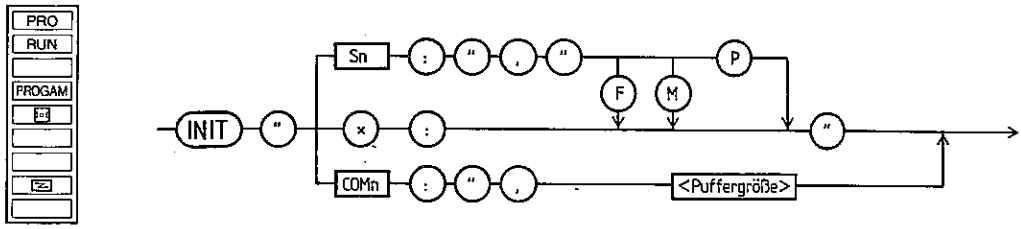
Bedingung ist erfüllt, wenn X den Wert 1 hat.

(P=2 AND Q=4) OR P=1

Die Bedingung ist erfüllt, wenn entweder P den Wert 1 besitzt (unabhängig von Q) oder aber P=2 und Q=4 gilt.

BEISPIEL :

```
10:INPUT "SOLL ICH PIEPSEN ",A$
20:IF A$="N" THEN 60
30:IF A$="J" THEN BEEP 3:GOTO 10
40:PRINT "NUR J ODER N EINGEBEN !"
50:GOTO 10
60:PRINT "SCHADE !"
70:END
```



Siehe auch : SETCOM, SETDEV, TITLE

WIRKUNG : INIT bewirkt folgendes:

1. Initialisiert die Module in den Modulfächern S1, S2 und spezifiziert ihren Gebrauch.
2. Initialisiert und formatiert Disketten.
3. Setzt die Größe des Empfangspuffers für die beiden seriellen Schnittstellen.

HINWEISE : **Modul-Initialisierung**

Das mit den Parametern S1: oder S2: ausgewählte Modul kann mit einem der folgenden Parameter für den gewünschten Einsatz initialisiert werden. (Dieses Befehlsformat ist nur in MODE 0 aktiv !)

"F" Formatiert das RAM-Modul als RAM-Disk, so daß auf diesem Dateien in gleicher Weise gespeichert und verwaltet werden können wie auf einer echten Diskette.

"M" Initialisiert das spezifizierte Modul als Erweiterung des Arbeitsspeichers, damit sich auch umfangreichere Programme in den Computer laden lassen.

"P" Initialisiert ein Modul als Programmspeicher, so daß sich auf diesem ein Programm auf unbestimmte Zeit speichern und jederzeit schnell zurückholen läßt. Wegen der im Modul befindlichen Pufferbatterie bleibt das Programm auch dann erhalten, wenn sich das Modul außerhalb des Computers befindet. Nach Einsetzung des Modules in den Computer ist dann das Programm sofort wieder nutzbar.

Der INIT-Befehl ist nicht anwendbar auf Module, die bereits Programme oder Dateien enthalten und mit dem Schreibschutzschalter geschützt sind (Schalterstellung ON).

Disketten-Initialisierung

INIT "X:" initialisiert und formatiert eine im Laufwerk befindliche Diskette. Neue Disketten müssen vor ihrem ersten Gebrauch formatiert werden, da sie sonst nicht benutzbar sind. Wird der Inhalt einer bereits gebrauchten Diskette nicht mehr benötigt, kann ihr gesamter Inhalt mit INIT gelöscht werden.

Empfangspuffer-Einstellung

Die Größe des Empfangspuffers für die Aufnahme der Daten, die über eine der beiden seriellen Schnittstellen (RS-232C, SIO) geliefert werden, wird mit der Anweisung:

INIT "COMn:" <Puffergröße> bestimmt.

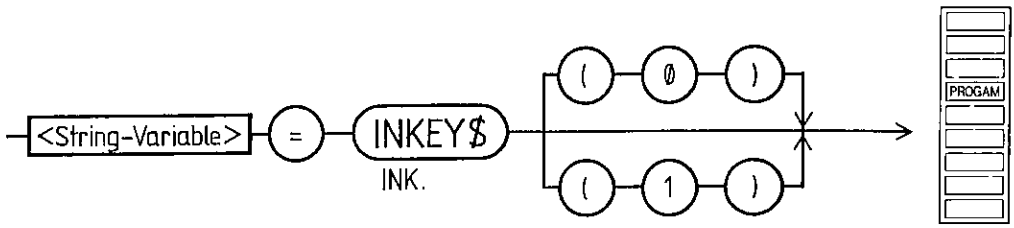
Der Puffer kann maximal 16383 Bytes groß sein und minimal zu 40 Bytes gewählt werden, wobei der Parameter **<Puffergröße>** jedoch im Bereich 80....16383 liegen muß. Die Puffergröße von 40 Bytes wird durch den Wert 0 eingestellt. Diese Größe wird nach jedem Einschalten des Computers automatisch als Standardwert eingestellt.

Mit dem Paramter **"COMn"** kann folgende Auswahl der Schnittstellen getroffen werden:

COM1:	RS-232C-Interface
COM2:	SIO-Interface
COM:	über SETDEV selektiertes Interface

Werden Dateien zwischen zwei Computern ausgetauscht, die direkt miteinander verbunden sind (ohne Modem, Akustikkoppler usw.), erlaubt ein großer Empfangspuffer einen schnellen Austausch. Werden keine Kommunikationsfunktionen benötigt, läßt andererseits ein minimaler Pufferspeicher (40 Bytes) einen umso größeren Arbeitsspeicher zu. In der Praxis muß also zwischen den beiden Extremen ein guter Kompromiß gefunden werden. Hierbei hat sich eine typische Puffergröße von 255 Bytes als ratsam ergeben.

Steht für die Einrichtung der spezifizierten Puffergröße nicht ausreichend Speicherplatz zur Verfügung oder befindet sich eine Datei für die APPEND-Option noch im geöffneten Zustand, wird ein ERROR-Code ausgegeben.



Siehe auch : INPUT

WIRKUNG : INKEY$ überprüft während der Programmausführung den Tastaturpuffer, ob zwischenzeitlich Zeichen über die Tastatur eingegeben worden sind, und liest ein einzelnes Zeichen in die spezifizierte Stringvariable ein.

HINWEISE : **INKEY$** und **INKEY$(0)** haben dieselbe Bedeutung und lesen jeweils das zuletzt über die Tastatur eingegebene Zeichen aus dem Tastaturpuffer und weisen es der spezifizierten Stringvariable zu.

Wurde zwischenzeitlich kein Zeichen über die Tastatur eingegegeben, liefert INKEY$ einen Nullstring (ASCII-Code 0).

Enthält der Tastaturpuffer mehrere Eingaben, kann mit **INKEY$(1)** die älteste Eingabe gelesen werden.

Der Unterschied zwischen den Befehlen INKEY$ und INPUT besteht darin, daß INPUT auf die Eingabe von Zeichen wartet und dieses dann auf dem Display anzeigt, während INKEY$ nicht wartet und auch keine Anzeige tätigt.

Mit der INKEY$-Anweisung lassen sich nur die Zeichen nachstehender Tabelle lesen:

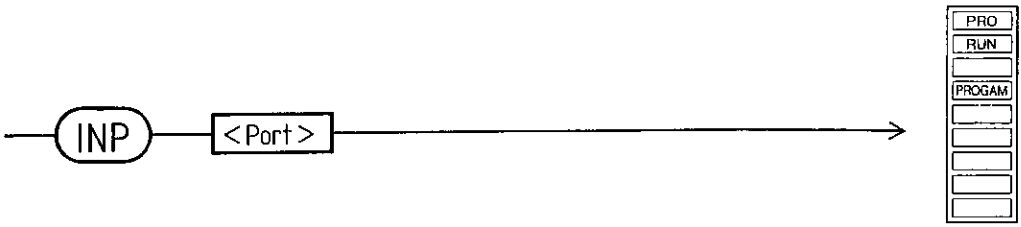
INKEY$ Character Table

Low \ High	0	1	2	3	4	5	6
0			SPACE	0		P	
1	SHIFT	F1		1	A	Q	
2	SML	F2		2	B	R	
3	CTRL	F3		3	C	S	
4	KBD/CLICK	F4		4	D	T	
5	BS	F5		5	E	U	
6		F6		6	F	V	
7				7	G	W	
8	◀	CL	(	8	H	X	
9	⬍	RCL	)	9	I	Y	
A	↓		*		J	Z	
B	↑	DEF	+		K		
C	▶				L		
D	ENTER		-	=	M		
E	ON		.		N		
F	OFF	MODE	/		O		

Abbildung F : Von INKEY$ akzeptierte Zeichen

BEISPIEL :

```
:
:
300:A$=INKEY$
310:IF A$="" THEN 300
320:IF A$="*" THEN 500
330:GOTO 300
:
500PRINT "HALLO"
:
```



Siehe auch : OUT

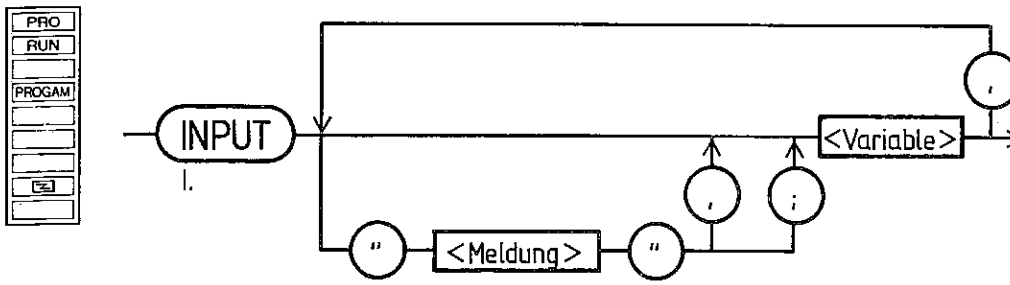
WIRKUNG : INP liefert ein Datenbyte vom spezifizierten Port des Z80-kompatiblen Mikroprozessors.

HINWEISE : Der Parameter <Port> bestimmt den Eingabe-Port, von dem ein Byte zu holen ist. Die Spezifikation des Ports geschieht durch eine Adresse, also mit mit einem 16-Bit-Wert im Bereich 0....65535 bzw. &0...&FFFF.

BEISPIEL :

```
:
300:A=INP(20)
:
```

INPUT



WIRKUNG : INPUT erlaubt während eines laufenden Programmes Variablen Werte zuweisen zu können.

HINWEISE : Mit Ausführung einer INPUT-Anweisung wird der Programmablauf gestoppt und eine <Meldung> auf dem Display angezeigt, sofern eine solche in der Anweisung spezifiziert worden ist. Fehlt diese, erscheint an ihrer Stelle ein Fragezeichen.

Während dieser Pause im Programmablauf können Daten über die Tastatur eingegeben oder über die seriellen Schnittstellen empfangen werden. Im letzten Fall müssen diese über SETDEV selektiert worden sein. Die entgegengenommenen Daten werden den als Parameter aufgelisteten <Variablen> der Reihe nach zugeordnet. Die Variablen sind in der Liste durch Kommas, die entsprechenden Tastatureingaben durch Betätigung der ENTER-Taste voneinander zu trennen.

Wird keine <Meldung> vereinbart, so erscheint in der Anzeige nur ein Fragezeichen, um auf eine notwendige Dateneingabe aufmerksam zu machen. Ist dagegen eine <Meldung> vereinbart, erfolgt vor Anzeige des Fragezeichens eben die Ausgabe dieser Meldung. Die Ausgabe des Fragezeichens läßt sich unterdrücken, wenn man dem Parameter <Meldung> ein Semikolon nachstellt.

In allen eben beschriebenen Fällen wird der Cursor hinter dem Fragezeichen bzw. der Meldung positioniert. Folgt der <Meldung> jedoch ein Komma, wird der Cursor auf den Anfang der Zeile gesetzt, die sich oberhalb der Zeile befindet, in der die Meldung angezeigt wird.

Um Daten von der RS-232C-Schnittstelle über den INPUT-Befehl einzulesen, ist das Format:

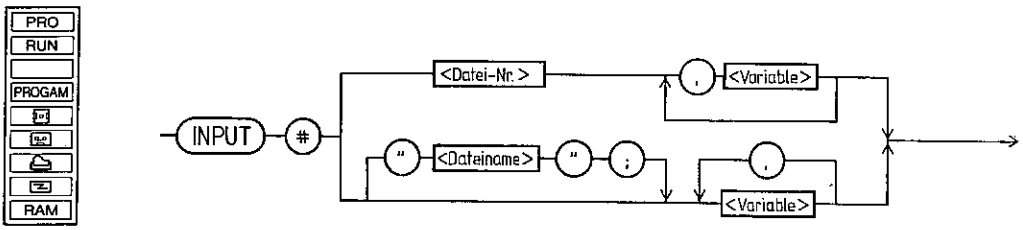
INPUT <Varible>, <Variable> ...

zu wählen. Hierbei kann weder eine <Meldung> vereinbart werden, noch erfolgt die Anzeige eines Fragezeichens.

BEISPIEL :

```
10:PRINT "BERECHNUNG: QUADER-VOLUMEN"
20:PRINT "Werte in Metern eingeben !"
30:INPUT "L=";L,"B=";B,"H=";H
40:V=L*B*H
50:PRINT "QUADER-VOLUMEN=";V;"m";CHR$(&FD)
60:END

>
RUN
BERECHNUNG: QUADER-VOLUMEN
Werte in Metern eingeben !
L=100
B=2*4
H=SQR(144)
QUADER-VOLUMEN= 9600m²
>
```



Siehe auch : DIM, INPUT, OPEN, PRINT#, SET

WIRKUNG : INPUT # liest Datensätze aus einer sequentiellen Datei, die sich auf einer Cassette, Diskette oder RAM-Disk befinden.

Disketten-Dateien und RAM-Disk-Dateien:

<Datei-Nr.> ist die Nummer, die der Datei bei ihrer Öffnung durch den OPEN-Befehl zugewiesen worden ist. Ein Versuch, eine ungeöffnete Datei zu lesen, endet mit der Ausgabe eines ERROR-Codes.

Cassetten-Dateien:

<Dateiname> bestimmt, welche Datei auf Cassette zu suchen und dann zu lesen ist. Fehlt dieser Parameter, wird die nächste auf Cassette befindliche Datei genommen.

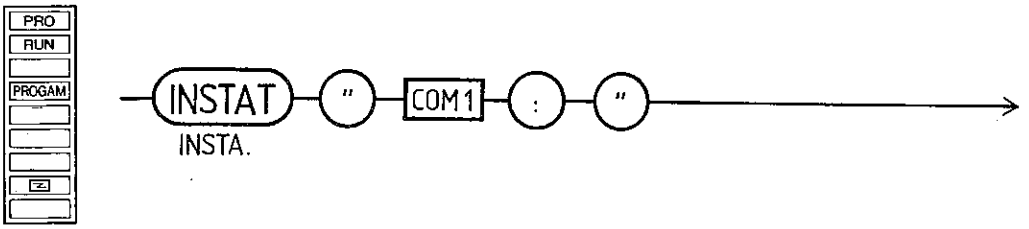
Cassetten-, Disketten- und RAM-Disk-Dateien:

<Variable> bestimmt die Namen der Variablen, in die die Daten der Datensätze einzulesen sind. Die Aufzählung der Variablen kann sowohl aus einfachen Variablen, Standard-Variablen oder Arrays bestehen. Reihenfolge und Typ der gelieferten Daten müssen zu den bereitgestellten Variablen passen. Stringvariable sind dabei in ausreichender Länge zu dimensionieren. Arrays müssen in der Variablen-Liste mit dem Pseudo-Index (*) versehen sein, z.B.: A(*).

Ist die Anzahl der gelesenen Daten geringer als die der aufgelisteten Variablen, hält das Programm an und wartet auf die restlichen Daten. Dieser Wartezustand kann durch Betätigung der BREAK Taste beendet werden. Ist die Anzahl der gelieferten Daten dagegegen größer als die der bereitgestellten Variablen, so werden alle überzähligen Daten ignoriert.

BEISPIEL :

```
10:MAXFILES=1
20:OPEN "X:=ADRESSE" FOR INPUT AS #1
30:PRINT "NEW YORK":PRINT
40:PRINT "NAME","TELEFON-NR."
50:IF EOF(1) THEN 100
60:INPUT #1,N$,S$,T$
70:IF S$="NEW YORK" THEN 90
80:GOTO 50
90:PRINT N$,T$ : GOTO 50
100:CLOSE #1
110:END
```



Siehe auch : OUTSTAT

WIRKUNG : INSTAT liefert einen numerischen Wert, der der Einstellung der RS-232C-Steuersignale angibt.

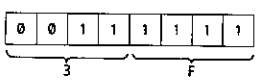
HINWEISE : Die Einstellung der Steuersignale wird durch einen 8-Bit-Wert, also einem Byte beschrieben. Die Zuordnung und Bedeutung ist wie folgt:

Bit	Signal	Bit=0	Bit=1
0	DTR	high	low
1	RTS	high	low
2	CTS	high	low
3	CD	high	low
4	DSR	high	low
5	CI	high	low

Die Bits 6 und 7 sind ständig 0.

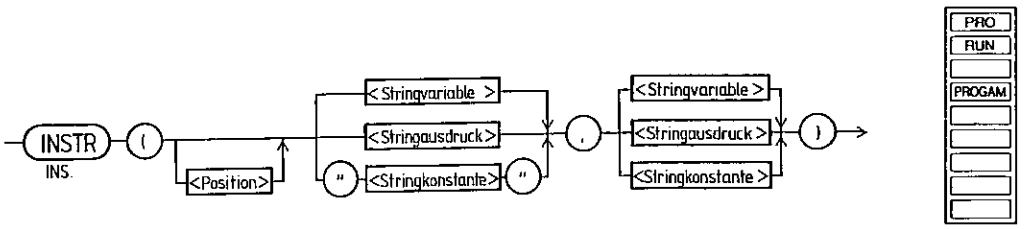
BEISPIEL :
```
INSTAT "COM1:"
 63
>
```

Der in diesem Beispiel gelieferte Wert kann in seiner Bedeutung nach der Umwandlung in den Hex-Wert (&3F) und dessen Umwandlung in die binäre Entsprechung wie folgt analysiert werden:



= 63 dezimal
= &3F hexadezimal
= 00111111 binär

Daraus folgt, daß alle sechs Steuersignale im Zustand "low" befindlich sind.



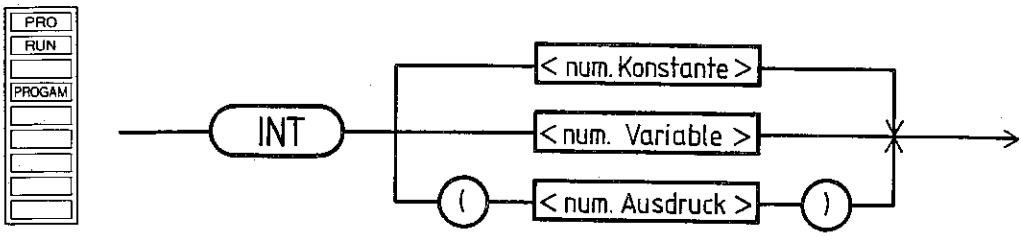
WIRKUNG : INSTR sucht in einem String nach dem erstmaligen Auftauchen eines Teilstringes und liefert dessen Position als Integer-Wert.

HINWEISE : Wird der Teilstring nicht gefunden oder ist der abzusuchende String ein Nullstring, so wird als Ergebnis der Wert 0 geliefert.

Der Parameter **<Position>** bestimmt, ab welcher Position mit der Suche zu beginnen ist. Fehlt diese Angabe, startet die Suche mit Position 1, also dem ersten Stringzeichen.

BEISPIEL :

```
10:INPUT "GIB EIN WORT EIN: ",W$
20:N=INSTR(W$,"A");
30:IF N=0 THEN 80
40:PRINT "DER BUCHSTABE A TAUCHT"
50:PRINT "ZUM ERSTEN MALE AN DER"
60:PRINT "POSITON ";N;" AUF."
70:GOTO 10
80:PRINT "IN DEM WORT KOMMT DER"
90:PRINT "BUCHSTABE A NICHT VOR !"
100:GOTO 10
```



WIRKUNG : Die Funktion INT(X) schneidet vom Argument X alle Nachkommastellen ab und liefert lediglich dessen ganzzahligen Anteil (INTeger-Wert).

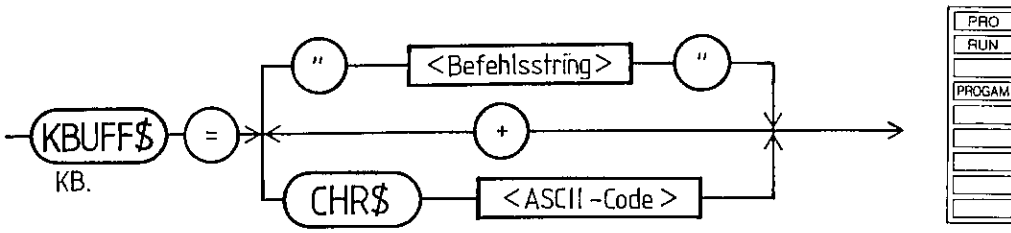
HINWEISE : Der Funktionswert ergibt sich aus der Abrundung des Argumentes auf den nächstniedrigeren ganzzahligen Wert.

Die Rundung ist völlig unabhängig davon, ob ein positiver oder negativer Wert als Argumentes angegeben ist. So werden positive Werte betragsmäßig kleiner, negative hingegen größer.

BEISPIEL :

```
10:PRINT INT(3.3)
20:PRINT INT(-3.3)
30:PRINT INT(.2)

>RUN
 3.
-4.
 0.
```



Siehe auch : INKEY$

WIRKUNG : KBUFF$ schreibt einen String von Zeichen in den Tastaturpuffer.

HINWEISE : Das KBUFF$-Kommando kann dazu verwendet werden, um den Computer im **"batch mode"** zu betreiben. Alle System-Kommandos können in der Reihenfolge ihrer gewünschten Abarbeitung in einer Datei abgelegt und später in den **<Befehlsstring>** eingelesen werden. KBUFF$ schreibt diesen String dann in den Tastaturpuffer, was die Wirkung hat, als ob die im String enthaltenen Kommandos über die Tastatur eingegeben worden seien.

Somit lassen sich automatisch Kommandos oder spezielle Aufgaben, wie z.B. Speicherlöschungen, das Laden spezieller Programme usw. ausführen.

Der Parameter **<Befehlsstring>** darf bis zu 32 Zeichen aufweisen. Dieser String überschreibt den gesamten bisher im Tastaturpuffer stehenden Inhalt.

BEISPIELE :

```
:
:
200:KBUFF$='45'
210:INPUT A
220:PRINT "A = ";A
:
:
```

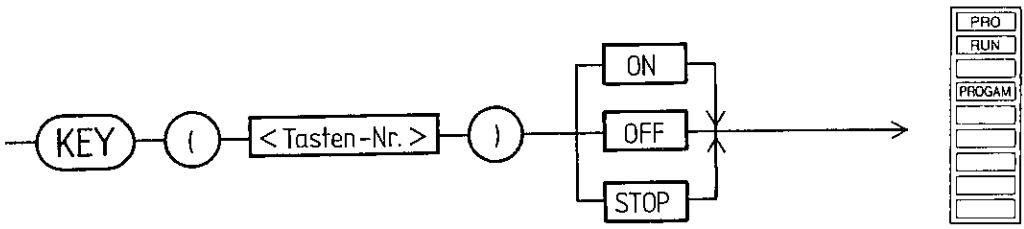
Zeile 200 schreibt die Zahl 45 in den Tastaturpuffer.

Zeile 210 wartet mit dem INPUT-Befehl auf die Eingabe eines numerischen Wertes. Da er bereits im Tastaturpuffer steht, nimmt der INPUT-Befehl diesen Wert, so als sei er gerade über die Tastatur eingetippt worden. Da mit dem KBUFF$-Kommando aber kein "carriage return" an den Puffer übergeben wurde, wartet die INPUT-Anweisung darauf, daß man die ENTER Taste betätigt.

Zeile 220 gibt zur Kontrolle den so eingelesenen Wert nochmals aus.

Um auch ein "carriage return" (Code &0D) oder andere nichtdarstellbare Zeichen in den Tastaturpuffer schreiben zu können, ist die Funktion CHR$ zu verwenden.

```
10:PAUSE "ENDLOSER SELBSTAUFRUF"
20:KBUFF$="RUN"+CHR$(&D)
```



Siehe auch : ON KEY GOSUB

WIRKUNG : Mit KEY ON/OFF/STOP kann die Verzweigung in eine Interrupt-Routine, die aufgrund der Betätigung der Funktionstasten gewünscht wird, verhindert oder erlaubt werden.

HINWEISE : Mit diesen Anweisungen kann also die Wirkung von ON KEY GOSUB beeinflußt werden.

KEY <Tasten-Nr.> OFF
hat zur Folge, daß in der Anweisung ON KEY GOSUB die genannte Funktionstaste unbeachtet bleibt.

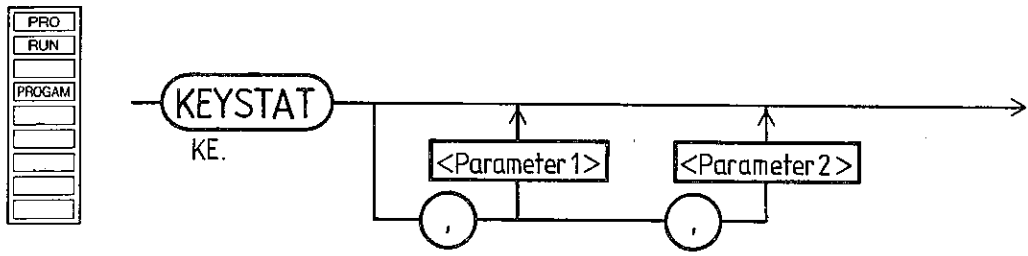
KEY <Tasten-Nr.> ON
bewirkt, daß die spezifizierte Taste von der Anweisung ON KEY GOSUB beachtet wird.

KEY <Tasten-Nr.> STOP
verhindert, wie bei der OFF-Option, ebenfalls das Wirksamwerden der angegebenen Funktionstaste im Zusammenhang mit der Anweisung ON KEY GOSUB. Sobald im Programm jedoch die Ausführung der Anweisung KEY <Tasten-Nr.> ON erfolgt, wird der bis dahin verweigerte Interrupt angenommen und ein Sprung gemäß der Verzweigungsvereinbarung durchgeführt. STOP ist die standardmäßig angenommene Option.

Eine Funktionstaste, die der Anweisung ON KEY GOSUB zugeordnet worden ist, kann nicht mehr als normale Funktionstaste dienen und vordefinierte Strings liefern.

BEISPIEL : siehe ON KEY GOSUB

KEYSTAT



Siehe auch : INKEY$

WIRKUNG : KEYSTAT bestimmt, woher die Eingabe von Befehlen standardmäßig erwartet wird, und ob ein Klickton zu hören ist, wenn man die Tasten betätigt. Ebenso bestimmt KEYSTAT, ob die Tasten eine Wiederholfunktion haben oder nicht.

HINWEISE : <Parameter 1> bestimmt die Wiederholfunktion:

0	Wiederholfunktion AUS
1	Wiederholfunktion AN

<Parameter 2> bestimmt die Klick-Funktion:

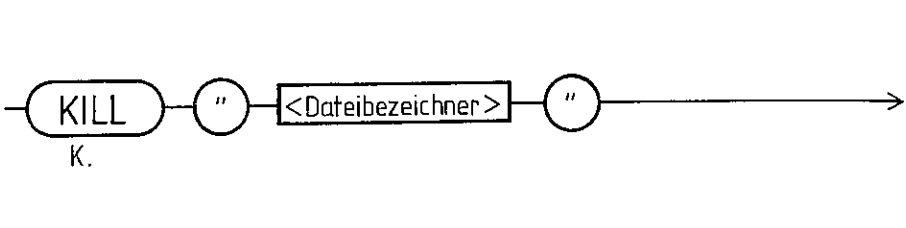
0	Klickfunktion AUS
1	Klickfunktion AN

Nach Einschaltung des Computers gilt standardmäßig die Einstellung:

KEYSTAT 0,0,0.

BEISPIEL :

```
>
KEYSTAT,1,1
```



Siehe auch : SAVE, SET

WIRKUNG : KILL löscht Dateien, die sich auf Diskette oder einer RAM-Disk befinden.

Der **<Dateibezeichner>** bestimmt welche Datei von welchem Speichermedium zu löschen ist. In diesem Bezeichner muß die Extension angegeben sein. Bei BASIC-Dateien ist also immer die Extension .BAS an den Dateinamen anzuhängen.

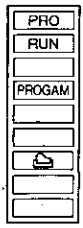
Die Benutzung von mehrdeutigen Namen unter Verwendung sogenannter "wildcards" (* oder ?) ist nicht möglich.

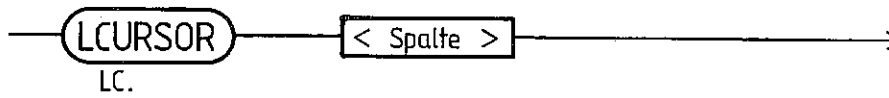
Ist die zu löschende Datei mit dem Attribut "P" (siehe SET-Befehl) geschützt, oder befindet sich der Schreibschutzschalter des RAM-Modules in der Stellung ON, wird ein ERROR-Code ausgegeben. Eine geöffnete Datei ist nicht löschbar und muß deshalb zuvor CLOSE geschlossen werden.

BEISPIEL :
```
>
KILL "S1:DATEI1"
```

Diese Anweisung löscht die Datei DATEI1 von der RAM-Disk des Modulfaches S1.

LCURSOR





Siehe auch : GLCURSOR, PCONSOLE, TAB

WIRKUNG : LCURSOR setzt den Druckstift an eine bestimmbare Text-Spalte, vorausgesetzt, der Drucker befindet sich im Text-Modus.

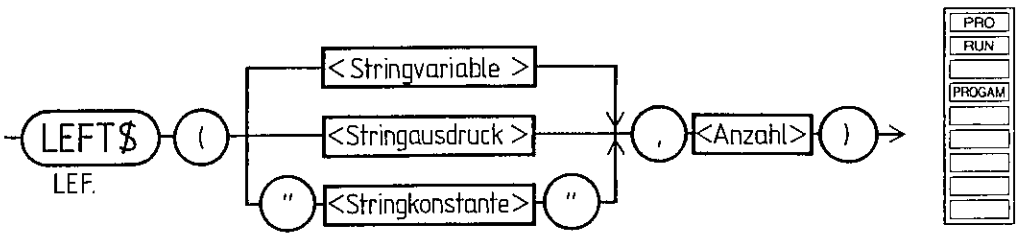
HINWEISE : Das LCURSOR-Kommando kann in seiner Wirkung mit dem GLCURSOR-Kommando verglichen werden, jedoch bewegt es den Druckstift nicht an jede beliebige X-Position, sondern nur an eine solche, bei der mit der Darstellung eines Zeichens im Text-Modus begonnen werden kann.

Der Parameter <Spalte> bestimmt also, an welche Zeichen-Position bzw. Text-Spalte der Druckstift gesetzt wird. Der Wert für diesen Parameter darf zwischen 0 und einem Maximalwert liegen, der um den Betrag 1 geringer ist als die über PCONSOLE spezifizierte maximale Zeilenlänge.

BEISPIEL :

```
10:LCURSOR 40
20:LPRINT "MITTE"
30:LPRINT "LINKS"
40:END
```

Dieses Programm läuft nur im Text-Modus.



Siehe auch : MID$, RIGHT$

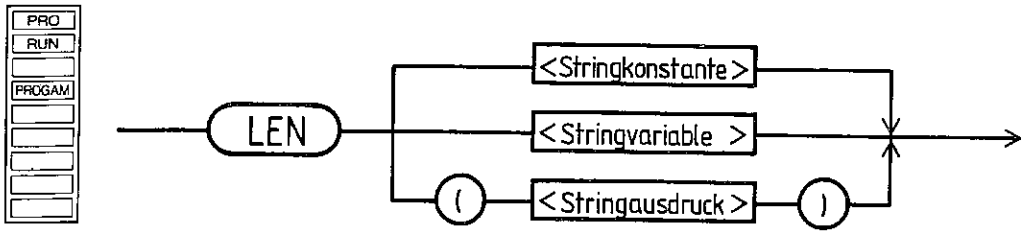
WIRKUNG : Die Funktion LEFT$ liefert den linksbündigen Teil eines vorgegebenen Strings, wobei bestimmt werden kann, wieviele Zeichen von rechts aus dem String abgelesen werden.

HINWEISE : Die **<Anzahl>** der Zeichen des Teilstringes muß im Bereich von 0 bis 80 liegen. Gibt man die Anzahl als gebrochene Zahl an, wird sie zur nächsten ganzen Zahl hin gerundet. Ist die Anzahl größer als die Zeichenanzahl des vorgegebenen Strings, wird der gesamte vorgegebene String geliefert.

BEISPIEL :

```
10:X$="SHARP PC-1600"
20:FOR N=1 TO 14
30:T$=LEFT$(X$,N)
40:PAUSE T$
50:NEXT N

>
RUN
S
SH
SHA
SHAR
SHARP
SHARP
SHARP P
SHARP PC
SHARP PC-
SHARP PC-1
SHARP PC-16
SHARP PC-160
SHARP PC-1600
SHARP PC-1600
```



WIRKUNG : LEN ermittelt die Länge eines Strings, d.h. die Anzahl der in ihm enthaltenen Zeichen.

ANMERKUNG : Diese Anzahl berücksichtigt auch solche Zeichen, die einem Leerzeichen (space) entsprechen oder aber zu den nicht darstellbaren Zeichen, wie den Steuercodes (control codes) gehören. Ein solches Steuerzeichen könnte z.B. ein "carriage return" (Sysmbol: <CR>, Code: &0D) sein.

BEISPIELE : Beispiel 1

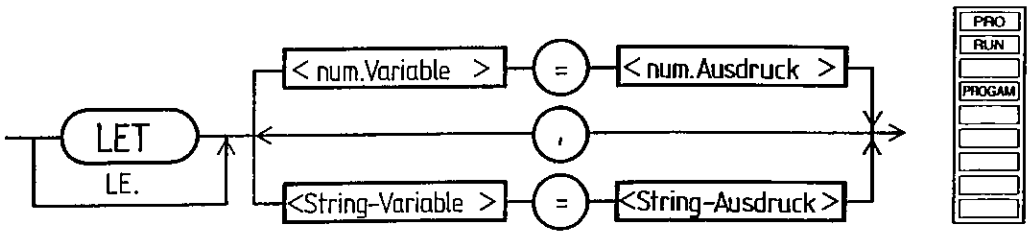
```
10:INPUT "GIB EIN WORT EIN :",W$
20:N=LEN(W$)
30:PRINT "DAS WORT HAT";N;" BUCHSTABEN"
40:END
```

Beachten Sie was passiert, wenn die Wörter aus mehr als 16 Zeichen bestehen.

Beispiel 2

```
10:A$="EINS";B$="ZWEI";C$="DREI"
20:S$=A$+CHR$(13)+B$+CHR$7+C$
30:PAUSE S$
40:PRINT "ANZAHL DER ZEICHEN = ";LEN S$
50:END

>
RUN
EINS ZWEI DREI
ANZAHL DER ZEICHEN =  14
>
```



WIRKUNG : LET weist Variablen Werte zu.

HINWEISE : Numerischen Variablen lassen sich nur numerische Werte zuweisen und Stringvariablen nur Strings.

Das Befehlswort LET ist optional und kann somit auch weggelassen werden. Damit sind die beiden folgenden Zuweisungen identisch:

LET A=5 oder einfach: A=5

Sind Wertzuweisungen hinter den Befehlswörtern THEN und ELSE erwünscht, so muß das Befehlswort LET verwendet werden, wenn eine solche Zuweisung direkt hinter dem Befehlswort erfolgt:

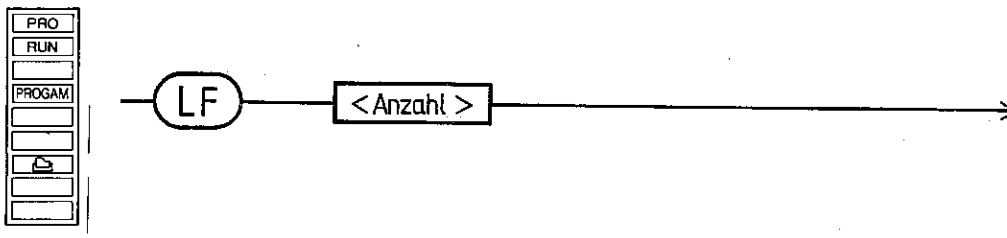
```
IF A=B THEN LET P=1 ELSE LET P=0
```

Erfolgt diese Zuweisung nicht direkt hinter dem Befehlswort THEN oder ELSE, weil sie an zweiter oder noch weiterer Stelle innerhalb einer Mehrfachanweisung steht, so kann LET entfallen.

```
IF A=B THEN BEEP 5: P=1 ELSE BEEP 1:P=0
```

BEISPIEL :

```
10:LET X$="1. STRING"
20:Y$="2. STRING"
30:PRINT X$: PRINT Y$
40:LET A=6
50:B=A+1
60:LET R$=LEFT$(X$,A),S$=LEFT$(Y$,B)
70:PRINT R$;S$
80:END
```



Siehe auch : CSIZE

WIRKUNG : LF bewegt das im Drucker eingespannte Papier um die gewünschte Zeilenanzahl vorwärts oder rückwärts.

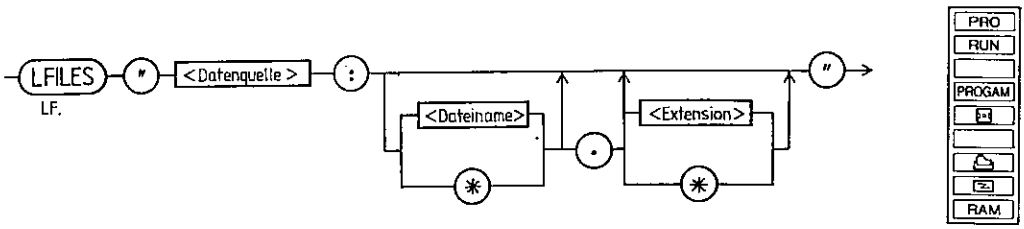
HINWEISE : Ohne Parameter bewegt das Kommando LF das Papier im Text-Modus um eine Zeile vorwärts.

<Anzahl> bestimmt, wenn dieser Parameter einen positiven Wert aufweist, um wieviele Zeilen das Papier vorwärtsbewegt wird. Ist dieser Wert negativ, bestimmt er entsprechend, um wieviele Zeilen das Papier rückwärts zu bewegen ist. Der Parameter <Anzahl> kann ein beliebiger numerischer Ausdruck sein. Von Werten, die nicht ganzzahlig sind, werden die Nachkommastellen abgeschnitten und nur der ganzzahlige Anteil berücksichtigt.

Der bei einer Vorwärts- oder Rückwärtsbewegung eingehaltene Zeilenabstand hängt von der mit CSIZE vorgenommenen Einstellung der Zeichengröße ab.

BEISPIEL :

```
10:LPRINT "   ZEILE"
20:LF 2
30:LPRINT "4. ZEILE"
40:LF-4
50:LPRINT "1."
60:LF(7)
70:END
```



Siehe auch : FILES, SETDEV

WIRKUNG : LFILES gibt ein Inhaltsverzeichnis der näher zu spezifizierenden Dateien eines Speichermediums über einen Drucker oder ein serielles Interface aus.

HINWEISE : Die Parameter von LFILES entsprechen denen vom Befehl FILES. Der Unterschied zu FILES besteht also nur darin, daß die Informationen nicht zum Display, sondern zum Drucker oder zu einer der beiden seriellen Ports geleitet werden.

Wohin die Daten geleitet werden, hängt von der über SETDEV spezifizierten Einstellung ab. Wurde SETDEV nicht benutzt, erfolgt standardmäßig die Ausgabe über den Drucker.

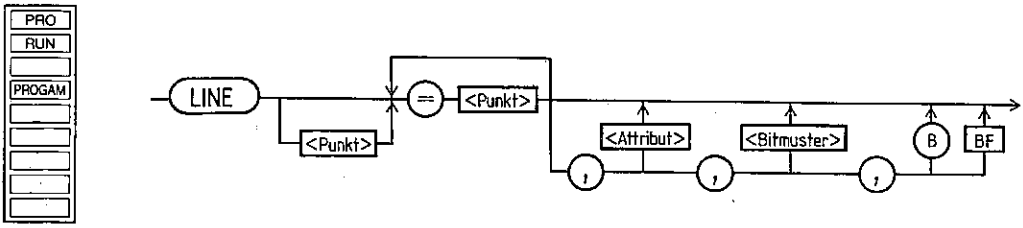
BEISPIELE :

```
>
LFILES "X:"
```

Diese Anweisung listet alle Dateien der mit X: angesprochenen Diskette auf.

```
>
LFILES "S2:*.BIN"
```

Diese Anweisung listet nur jene Dateien der mit S2: selektierten RAM-Disk(ette) auf, die mit der Extension BIN behaftet sind.



Siehe auch : LLINE

WIRKUNG : LINE zeichnet eine Gerade (Linie) zwischen zwei Display-Punkten.

Die gezeichnete Linie erstreckt sich dabei vom 1. **<Punkt>** mit den Koordinaten (X1,Y1) bis zum 2. **<Punkt>** mit den Koordinaten (X2,Y2) des als Punktmatrix aufgefaßten Displays.

Der Ursprung (0,0) des zugrundegelegten Koordinatensystemes befindet sich dabei in der linken oberen Display-Ecke.

Wird die Angabe des 1. Punktes weggelassen, so wird hierfür die momentane Position des Grafik-Cursors angenommen.

<Attribut> bestimmt, welchen Einfluß die Linie auf die Display-Punkte hat, die von dieser berührt werden:

S Alle im **<Bitmuster>** gesetzten Bits steuern die berührten Display-Punkte dunkel, alle gelöschten Bits steuern diese hell.

R Alle im **<Bitmuster>** gesetzten Bits steuern die berührten Display-Punkte hell, alle gelöschten Bits steuern diese dunkel. Damit wirkt R also in entgegengesetzter Weise zu S.

X Invertiert alle von der Verbindungslinie berührten Display-Punkte.

Fehlt die Angabe des Attributes, wird für dieses standardmäßig die Option S angenommen.

<Bitmuster> ist ein 16-Bit-Wert, liegt also im Bereich von 0 bis 65535 und bestimmt das Aussehen der Linie.

Beispiele:

65535 (&FFFF) liefert eine durchgehende Linie:

Binär-Wert von &FFFF = 1111111111111111
Die Linienstruktur ist: xxxxxxxxxxxxxxxx

43690 (&AAAA) liefert eine gepunktete Linie:

Binär-Wert von &AAAA = 1010101010101010
Die Linienstruktur ist: x x x x x x x x

26214 (&6666) liefert eine gestrichelte Linie:

Binär-Wert von &6666 = 0110011001100110
Die Linienstruktur ist: xx xx xx xx

Fehlt der Parameter <Bitmuster>,wird eine durchgehende Linie gezeichnet.

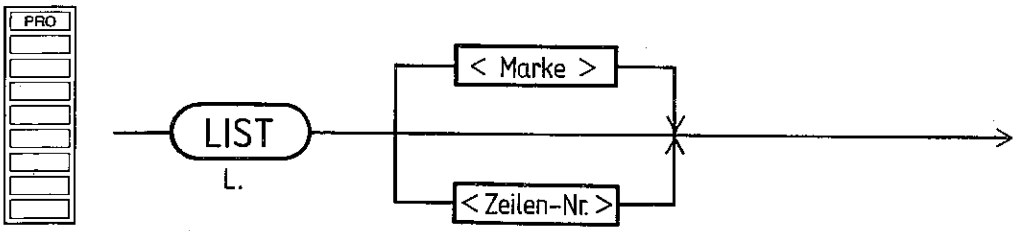
B Zeichnet ein Rechteck, wobei die angegebenen Punkte als diagonal gegenüberliegende Eckpunkte des Rechtecks angesehen werden.

BF Zeichnet ebenfalls ein Rechteck, wie bei der Option B, füllt es jedoch mi dem <Bitmuster> aus.

Die X- und Y-Koordinaten eines Punktes können im Bereich -32768....32767 angegeben werden, obwohl nur die Werte 0...155 für X und 0...31 für Y im sichtbaren Bereich des Displays liegen.

BEISPIEL :

```
10:CLS
20:FOR N=10 TO 100 STEP 30
30:M=N+20
40:LINE (N,10)-(M,20),,,BF
50:NEXT N
60:END
```



Siehe auch : LLIST

WIRKUNG : LIST zeigt die Zeilen eines BASIC-Programmes auf dem Display an.

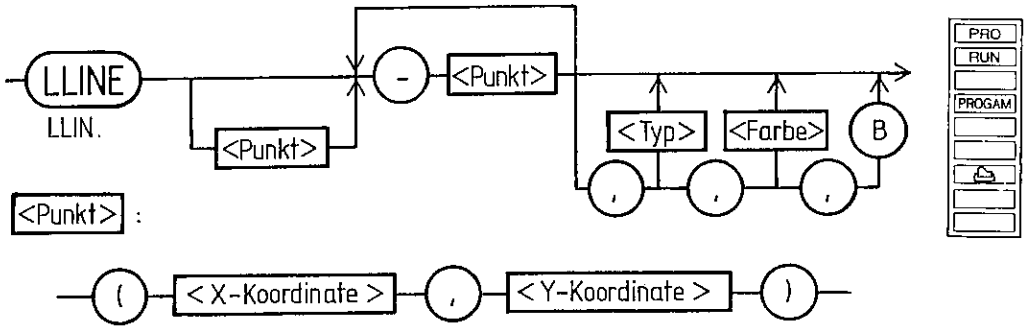
HINWEISE : Ohne Angabe einer <Zeilen-Nr.> beginnt LIST die zu erstellende Auflistung mit der ersten im Programm enthaltenen Zeile. Sie wird in der oberen Display-Zeile dargestellt. Soweit es der Platz zuläßt, zeigt das Display auch die nachfolgenden Programmzeilen an. Der Cursor wird unsichtbar hinter der ersten Zeilennummer positioniert und durch einen blinkenden Doppelpunkt kenntlich gemacht.

Alle weiteren Zeilen können dadurch zur Anzeige gebracht werden, indem man mit der Taste [↓] den Cursor nach unten bewegt und den Display-Inhalt nach oben hin wegrollen (scrollen) läßt.

Wird LIST der Parameter **<Zeilen-Nr.>** beigefügt, so beginnt die Auflistung eben mit dieser Zeile. Existiert im Programm keine Zeile dieser Nummer, so wird die Auflistung mit der Zeile begonnen, die die nächsthöhere Nummer aufweist. Liegt die <Zeilen-Nr.> über der höchsten im Programm vorkommenden Zeilennummer, so wird ein ERROR-Code angezeigt. Anstelle der <Zeilen-Nr.> ist auch eine existierende **<Marke>** angebbar.

Ist das Programm mit der "P"-Option des Befehles SET geschützt, wird bei einem LIST-Versuch ebenfalls ein ERROR-Code generiert.

Ein Programm, das über PASS mit einem Kennwort geschützt ist, läßt sich nicht auflisten, da der Zugang zum PRO-Modus in diesem Falle verwehrt ist, der LIST-Befehl aber nur in diesem Modus akzeptiert wird.

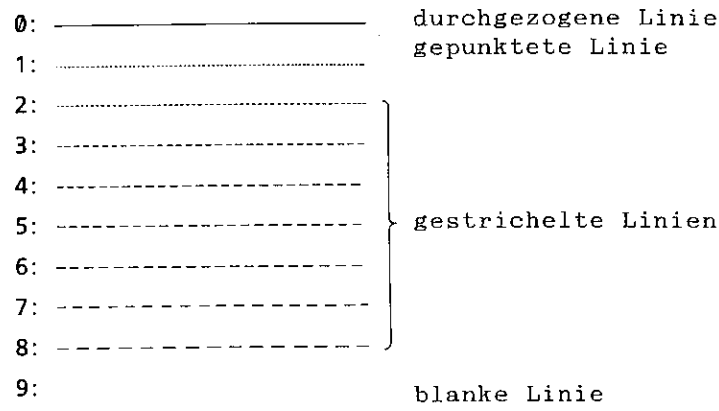


Siehe auch : COLOR, RLINE, SORGN

WIRKUNG : LLINE zeichnet eine oder eine Reihe von Linien zwischen den jeweils definierten Punkten des im Drucker eingespannten Papieres.

HINWEISE : Im Grafik-Modus zeichnet LLINE eine vom **<Punkt>** (X1,Y1) (oder wenn diese Angabe fehlt: von der momentanen Stiftposition) ausgehende Gerade zum <Punkt> (X2,Y2). Die X- und Y-Koordinaten müssen jeweils im Bereich -2048...2047 liegen. Sie sind absolut, d.h., sie beziehen sich auf den mittels SORGN festgelegten Koordinatenursprung (0,0). In der LLINE-Anweisung können bis zu fünf zusammenhängende Linien vereinbart werden.

<Typ> bestimmt die aus 9 verschiedenen Mustern gewünschte Linienstruktur:



Der <Typ> 9 dient zum Test der Stiftbewegung.

Fehlt der Parameter <Typ>, bleibt die zuletzt vorgenommene Einstellung erhalten.

Wird LLINE unmittelbar nach einem LPRINT-Befehl im Grafik-Modus ausgeführt, ist in diesem Falle der Parameter <Typ> unbedingt mit anzugeben, da hier kein Standardwert angenommen wird. Dieses betrifft auch übernommene PC-1500-Programme, die LINE-Anweisungen enthalten.

<Farbe> bestimmt den zu benutzenden Farbstift. Es stehen vier Farbstifte zur Auswahl:

0	schwarz
1	blau
2	grün
3	rot

Bei fehlender Angabe des Parameters <Farbe> wird der momentan selektierte Stift genommen.

Die Angabe der Option B sorgt für das Zeichnen eines Rechteckes, das die beiden Punkte (X1,Y1) und (X2,Y2) als Diagonale hat.

Die Anweisung LLINE (X1,Y1)-(X2,Y2),,,B ergibt also folgendes Bild:

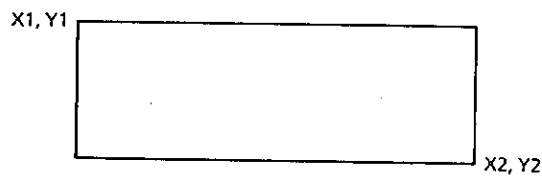
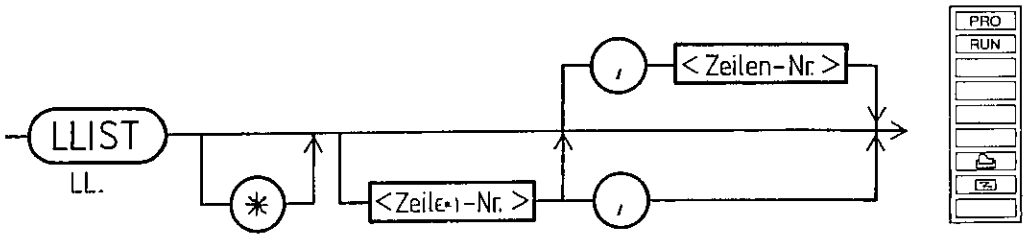


Abbildung 42

BEISPIEL :

```
10:GRAPH
20:GLCURSOR (40,40)
30:SORGN
40:LLINE -(100,0)-(0,100)-(0,0)
50:TEXT
60:END
```



Siehe auch : LIST

WIRKUNG : LLIST listet die im Speicher befindlichen Zeilen eines BASIC-Programmes mit Hilfe eines Druckers auf Papier auf oder sendet diese über eines der beiden seriellen Ports aus.

LLIST wird zwar in ähnlicher Weise verwendet wie der Befehl LIST, ist jedoch flexibler in seinen möglichen Parameterangaben.

Ausgabe zum Drucker:

LLIST
sendet das komplette Programm, also alle Zeilen des Programmes zum Drucker.

LLIST <Zeilen-Nr.>
bringt nur die gewünschte Zeile zu Papier.

LLIST <Zeilen-Nr.>,<Zeilen-Nr.>
beginnt die Auflistung mit der Zeile der zuerst angegebenen Zeilennummer und beendet diese mit der Zeile der als zweites angegebenen Zeilennummer.

LLIST <Zeilen-Nr.>,
beginnt die Auflistung mit der spezifizierten Zeile und setzt diese bis hin zum Programmende fort.

LLIST ,<Zeilen-Nr.>
beginnt die Auflistung mit der ersten Programmzeile und beendet sie mit Ausdruck der angegebenen Zeile.

Wird LLIST im Zusammenhang mit dem Drucker verwendet, so berücksichtigt dieser die Einstellung der mit PCONSOLE bestimmten Zeilenlänge. Ist sie größer oder gleich 18 Zeichen gesetzt, werden alle Zeilen, die über diese Länge hinausgehen, in der nächsten Druckzeile fortgesetzt. Ist sie jedoch zu 16 oder 17 bestimmt worden, erzeugen alle längeren Zeilen die Anzeige von ERROR 76. In diesem Falle unterbleibt der Ausdruck der betreffenden Programmzeile.

Ist mit CSIZE die <Größe> der Zeichen mit dem Wert 1 festgelegt, so erfolgt der Ausdruck in dieser Zeichengröße. Bei allen anderen Werten werden die Zeichen in der Größe abgebildet, als sei CSIZE 2 eingestellt worden.

LLIST und LLIST* schalten automatisch in den Text-Modus um.

Vor dem Ausdruck jeder Zeile wird der mit dem Befehl PCONSOLE bestimmte <Heftrand> beachtet.

Ausgabe über einen seriellen Port:

Ist mit dem Befehl SETDEV und der Option PO ein serieller Port für eine Datenausgabe geöffnet worden, sendet LLIST alle Programmzeilen nun zu dem betreffenden Port anstatt zum Drucker.

Jede Zeile enthält die mit PCONSOLE festgelegte Zeichenanzahl und wird mit dem von PCONSOLE bestimmten "end of line code" beendet.

Ausgabe über Drucker und Interface:

Ist das Programm mit einem Kennwort mittels PASS geschützt, werden LLIST und LLIST* ignoriert.

LLIST* listet nur diejenigen Programmzeilen auf, die mit einem Apostroph ' beginnen. Alle Kommentare, die mit dem Befehlswort REM gekennzeichnet sind oder nicht am Zeilenanfang stehen, werden von LLIST* nicht berücksichtigt.

Mit LLIST* lassen sich alle wichtigen in einem Programm befindlichen Kommentare herausfiltern. Endet eine mit Apostroph beginnende Kommentarzeile mit einem Semikolon, so wird der nächste Kommentar in derselben Druckzeile angefügt.

Benutzt man LLIST* im Zusammenhang mit einem seriellen Port, so werden alle Programmzeilen, die ab 100 aufwärts numeriert sind, so in ihrem Inhalt abgeschnitten, daß sie in den Puffer passen. Um unerwünschte Beschneidungen zu vermeiden, sollten deshalb nur Zeilennummern bis einschließlich 99 verwendet werden.

BEISPIEL :

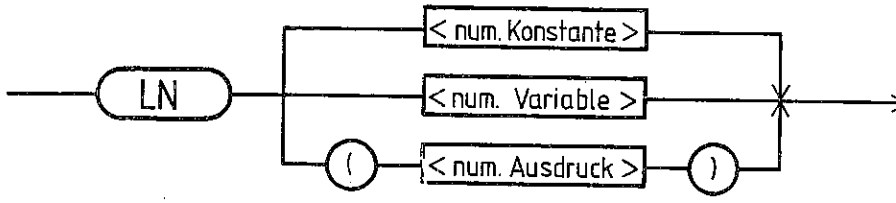
```
10:REM TEST-PROGRAMM
20:'1. KOMMENTAR ;
30:PRINT "HALLO, ";
40:'2. KOMENTAR
50:PRINT "HIER  "; 'INGNORIERTER KOMMENTAR
60:PRINT "IST DER PC-1600 !"
70:REM ENDE DES PROGRAMMES
80:END

>
LLIST 20,60

20:'1. KOMMENTAR ;
30:PRINT "HALLO, ";
40:'2. KOMENTAR
50:PRINT "HIER "; 'INGNORIERTER KOMMENTAR
60:PRINT "IST DER PC-1600 !"


>
LLIST*

1. KOMMENTAR 2. KOMMENTAR
```



WIRKUNG : Die Funktion LN(X) liefert den natürlichen Logarithmus des Argumentes X.

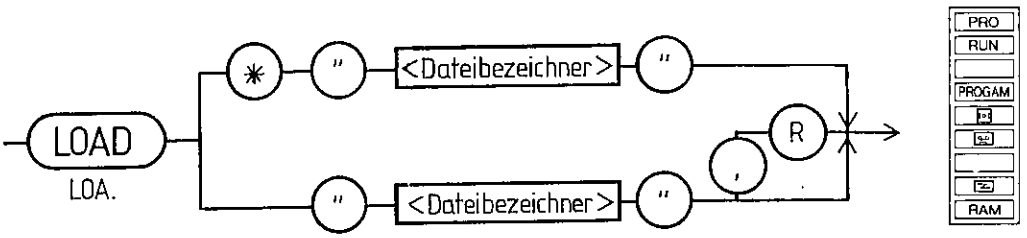
Die Basis dieser Lograrithmus-Funktion ist die Zahl e. Die Funktion LN ist die Umkehrung der Funktion EXP.

Als Argument ist jeder beliebige numerische Ausdruck erlaubt, sofern sein Resultat innerhalb des zulässigen Wertebereiches liegt.

Das Argument muß größer oder gleich 1E-99 sein. Werte, die darunter liegen, bewirken die Anzeige des ERROR-Codes 39.

BEISPIEL :

```
10:CLS: INPUT "Argument = ";X
20:PRINT "Der Logarithmus zur Basis"
30:PRINT "e lautet: ";LN(X)
40:INPUT "Weitere Berechnung (J/N)";A$
50:IF A$="J" THEN 10
60:END
```



Siehe auch : CHAIN, LLIST*, MERGE, REM, RUN, SAVE

WIRKUNG : LOAD lädt eine Datei in den internen Speicher, die sich auf einer Cassette, Diskette oder RAM-Disk befindet oder aber über einen der beiden seriellen Ports geliefert wird.

HINWEISE : Im <Dateibezeichner> kann die <Datenquelle> aus folgenden Medien auswählen:

S1:	RAM-Disk des Modulfaches S2
S2:	RAM-Disk des Modulfaches S2
X:, Y:	Diskette
COM1:	RS-232C-Interface
COM2:	SIO-Interface
CAS:	Cassette

Ist dem Ladebefehl LOAD die Option R beigefügt, wird die Datei geladen und das in ihr enthaltene BASIC-Programm automatisch gestartet. Besteht die Datei nicht aus einem Programm, erfolgt die Anzeige eines ERROR-Codes.

Alle offenen Dateien, die nicht zur mit LOAD geladenen gehören, werden durch LOAD geschlossen, sofern nicht die R-Option angegegben ist.

LOAD* setzt an den Beginn jeder geladenen Zeile eine Zeilennummer mit nachfolgendem Apostroph. Dies hat den Effekt, daß im Speicher ein BASIC-Programm entsteht, das sich ausschließlich aus Kommentarzeilen zusammensetzt. Die Numerierung der Zeilen beginnt mit 10 und wird im Zehnerabstand fortgesetzt.

Durch Anwendung von LOAD* ist es möglich, ASCII-Dateien in den Arbeitsspeicher zu laden. Dieses ist der einzige Weg, mit dem PC-1600 auf dieser Ebene Textdateien zu verarbeiten. Eine solche Textdatei kann mit LLIST* ausgedruckt werden.

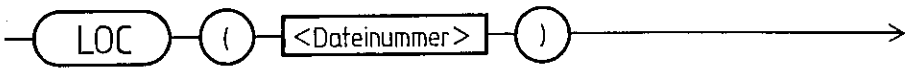
Eine Datei mit folgenden Text wird mittels LOAD* so im Speicher abgelegt:

```
10 ' LOAD* ERLAUBT ES, AUF
20 ' BASIC-EBENE TEXTE ZU
30 ' LADEN UND AUF DEM
40 ' DISPLAY SICHTBAR ZU
50 ' MACHEN. DIES REICHT,
60 ' UM EINFACHE NOTIZEN
70 ' ODER ADRESSEN AUFNEH-
80 ' MEN ZU KOENNEN.
```

BEISPIEL :

```
>
LOAD "BIOCALC",R
```

Diese Anweisung lädt von der momentan gültigen Datenquelle das Programm BIOCALC und startet es anschließend automatisch.



WIRKUNG : LOC gibt die Anzahl der bisher gelesenen oder geschriebenen Datensätze der mit <Dateinummer> spezifizierten Datei an.

HINWEISE : LOC ist nur im Zusammenhang mit Dateien verwendbar, die sich auf einer Diskette oder einer RAM-Disk befinden.

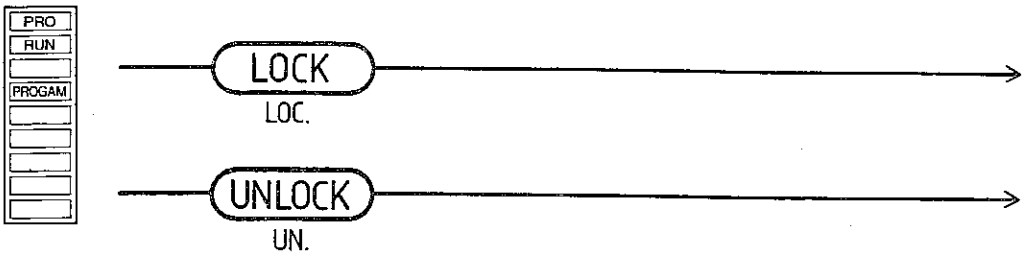
Mit LOC kann der Programmablauf in Abhängigkeit von der Anzahl der gelesenen oder geschriebenen Datensätze gesteuert werden, solange die Datei geöffnet ist.

BEISPIEL :

```
10:OPEN "X:DATEI1" FOR INPUT AS #1
20:IF EOF(1) THEN 50
30:INPUT #1,N
40:GOTO 20
50:M=LOC(1)
60:PRINT "DIE DATEI BESTEHT AUS"
70:PRINT M;" DATENSÄTZEN"
80:CLOSE #1
90:END
```

Zeile 10 öffnete die Datei DATEI1 zum Lesen.
Zeile 20 überprüft, ob das Dateiende erreicht ist und verzweigt gegebenenfalls zu Zeile 50.
Zeile 30 liest einen Datensatz, der hier nur aus einem numerischen Wert besteht.
Zeile 40 bereitet einen weiteren Lesevorgang vor.
Zeile 50 übergibt an die Variable M die Anzahl der insgesamt gelesenen Datensätze.
Zeile 70 zeigt diese Anzahl an und
Zeile 80 schließt die gelesene Datei.
Zeile 90 beendet das Programm.

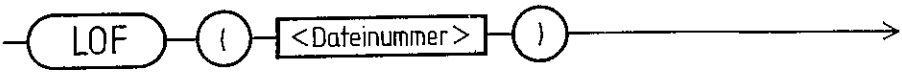
LOCK / UNLOCK



WIRKUNG : LOCK schaltet die Funktionsfähigkeit der Taste MODE ab. UNLOCK schaltet sie wieder ein.

HINWEISE : Mit LOCK kann verhindert werden, daß durch die versehentliche Berührung der MODE-Taste in einen verkehrten Betriebsmodus geschaltet wird. Trotz Betätigung dieser Taste behält der Computer den derzeit aktiven Modus bei.

Mit UNLOCK kann die MODE-Taste wieder funktionsfähig gemacht und der Computer in einen anderen Modus geschaltet werden.



WIRKUNG : LOF gibt die Anzahl der Bytes an, aus denen eine Datei besteht.

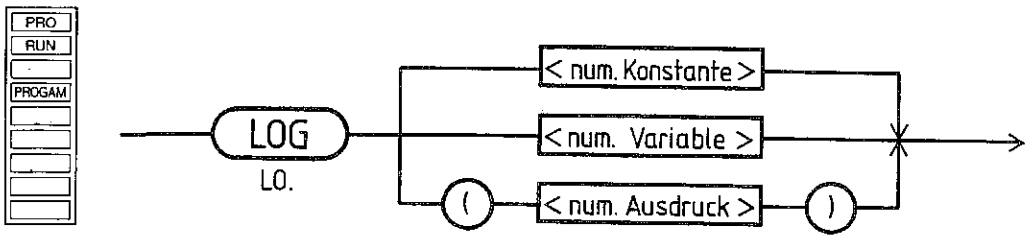
HINWEISE : LOF ermittelt die Größe von Dateien, die sich entweder auf einer Diskette oder einer RAM-Disk befinden.

Dazu muß die Datei über den OPEN-Befehl geöffnet und von diesem mit einer **<Dateinummer>** versehen worden sein.

Über diese **<Dateinummer>** ist die Datei, deren Größe ermittelt werden soll, anzusprechen.

BEISPIEL :

```
10:OPEN "X:DATEI1" FOR INPUT AS #1
20:N=LOF(1)
30:PRINT "DATEI1 BESTEHT AUS ";N
40:PRINT "BYTES."
50:CLOSE #1
60:END
```



Siehe auch : LN

WIRKUNG : Die Funktion LOG liefert den dekadischen Logarithmus des angegebenen Argumentes.

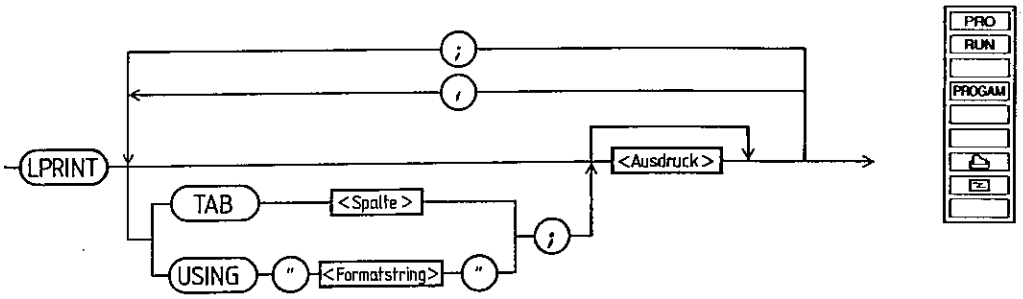
HINWEISE : Um einen Logarithmus zu einer anderen Basis als 10 zu erhalten, z.B. zur beliebigen Basis B, ist die folgende Formel zu verwenden:

```
LOG(X)/LOG(B)
```

Die Umkehrung des Zehnerlogarithmus kann mit der Hilfe des Potenzoperators ^ gebildet werden, wenn man als Potenzbasis die Zahl 10 wählt.

BEISPIEL :

```
>LOG (2)
                3.010299957E-01
>
```



Siehe auch : PCONSOLE, PRINT, PZONE, TAB

WIRKUNG : LPRINT gibt Daten über einen Drucker oder über eine serielle Schnittstelle aus.

Die Anweisungen LPRINT und LPRINT USING werden dabei in gleicher Weise verwendet wie PRINT und PRINT USING.

Ist über SETDEV ein serieller Port selektiert, gibt LPRINT die auszugebenden Daten nicht an den Drucker, sondern an dieses Interface weiter.

Wenn im folgenden einmal nur von dem Drucker die Rede ist, denken Sie bitte daran, daß die Datenausgabe auch für die Ports gelten kann, obwohl wenn es nicht ausdrücklich hervorgehoben ist.

LPRINT ohne Parameter sorgt für den Vorschub des Papieres um eine Zeile.

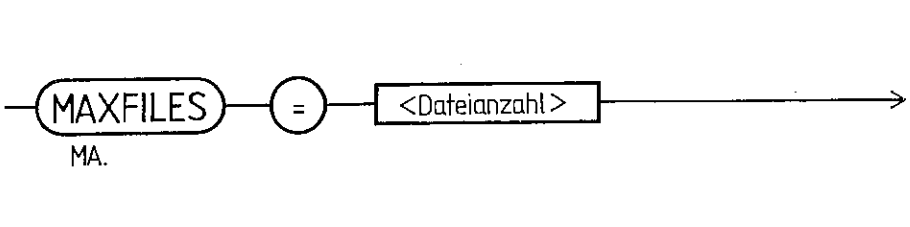
LPRINT mit Parameterangabe sendet die Werte der aufgelisteten Ausdrücke nacheinander aus. Diese Ausdrücke können sowohl numerisch sein oder auch einem String entsprechen. Wird ein Semikolon zur Trennung der Ausdrücke verwendet, so werden ihre Werte unmittelbar hintereinander ausgegeben. Ist als Trennzeichen ein Komma gesetzt, so wird der Wert des nachfolgenden Ausdruckes in die nächste durch PZONE bestimmte Position gedruckt.

Endet die Liste der Ausdrücke mit einem Semikolon, wird die folgende LPRINT-Anweisung an der nächsten mit PZONE vereinbarten Position fortgesetzt. Schließt kein Semikolon die besagte Liste ab, wird ein Zeilenvorschub ausgegeben.

LPRINT USING verhält sich in Analogie zu PRINT USING, so daß Sie hierzu auf die dort gemachten Beschreibungen zurückgreifen können.

Die TAB-Option spezifiziert, in welcher Spalte der Druck der nächsten Ausgabe erfolgen soll. Überschreitet die Spaltenangabe den Wert der mit PCONSOLE festgelegten Zeilenlänge, erfolgt die Anzeige eines ERROR-Codes.

Bei Verwendung von LPRINT im Grafik-Modus wird kein Zeilenende-Code (EOL-Code) wie <CR> (&0D) oder <LF> (&0A) am Zeilenende ausgegeben.



Siehe auch : CLOSE, OPEN

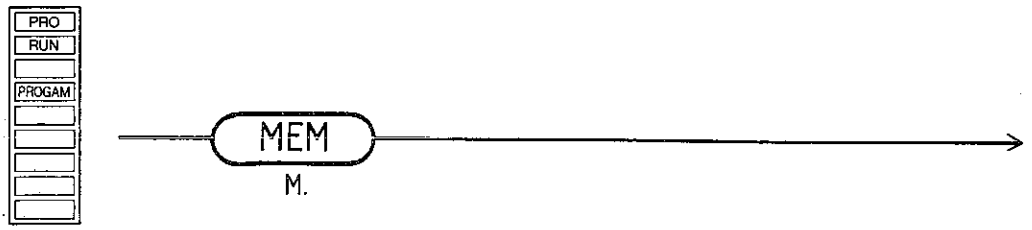
WIRKUNG : MAXFILES bestimmt die Anzahl der Dateien, die gleichzeitig geöffnet sein dürfen.

HINWEISE : Der maximale Wert für **<Dateianzahl>** beträgt 15.

Alle Dateien müssen geschlossen sein, bevor das Kommando MAXFILES ausgeführt werden kann. Nach Einschaltung des Computers ist diese Anzahl automatisch zu Null gesetzt. Somit muß MAXFILES nach Einschaltung des Gerätes mindestens einmal aktiviert werden, damit überhaupt ein Zugriff auf Dateien erfolgen kann.

Für jede zu verwaltende Datei werden durch den Befehl MAXFILES 313 Bytes reserviert. Ist nicht genügend Speicherplatz frei, um diese Bereiche für die Verwaltung anlegen zu können, wird ein ERROR-Code angezeigt.

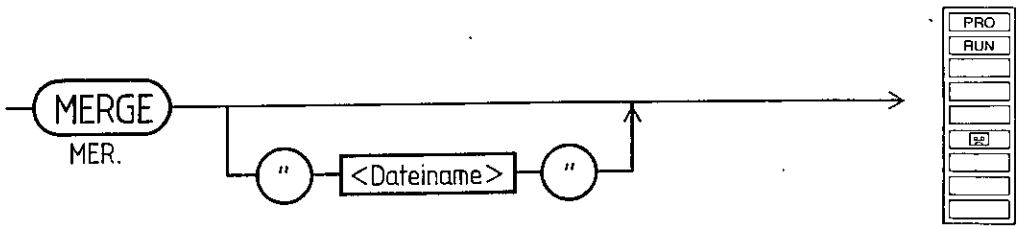
BEISPIEL :
```
>
MAXFILES=3
```



Siehe auch : STATUS

WIRKUNG : MEM liefert die Anzahl der noch freien Bytes des Arbeitsspeichers.

HINWEISE : Die gelieferte Anzahl schließt dabei den Bereich der Variablen ein. Damit ist MEM gleichbedeutend mit der Anweisung STATUS 0.



Siehe auch : CLOAD

WIRKUNG : MERGE lädt ein weiteres BASIC-Programm von der Cassette in den Arbeitsspeicher ohne das bereits vorhandene Programm zu löschen.

HINWEISE : Wird MERGE ohne Parameterangabe verwendet, lädt der PC-1600 das nächste Programm, das er auf der Cassette vorfindet. Mit Parameterangabe wird das mit **<Dateiname>** bestimmte Programm von Cassette in den Speicher geholt.

Das mit MERGE geladene Programm wird an das Ende des bereits im Speicher befindlichen Programmes angefügt. Es darf sogar Zeilennummern aufweisen, die auch in dem schon im Speicher stehenden Programm vorkommen. Dieses ist möglich, weil nach Ausführung von MERGE grundsätzlich nur auf das zuletzt geladene Programm mit RUN und LIST zugegriffen werden kann. Deshalb sollte man alle Programme mit einer Markierung versehen, um mit GOTO <Marke> und LIST <Marke> das gewünschte Programm ansprechen zu können.

Bei Verwendung von MERGE sollten auch die DATA-Anweisungen mit einer Markierung versehen sein, damit die Read-Anweisung innerhalb des richtigen Programmes zugreift. Um dieses sicherzustellen, muß vor dem ersten READ ein RESTORE-Befehl ausgeführt werden, der sich auf diese Markierung bezieht. Beispiel:

```
10:RSTORE "A"
20:READ X,Y,Z
30:PRINT X;Y;Z
40:"A":DATA 3,4,5
```

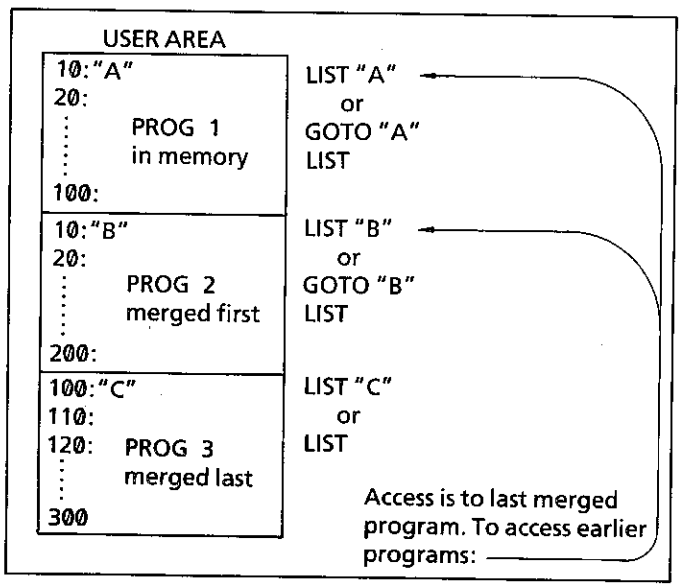
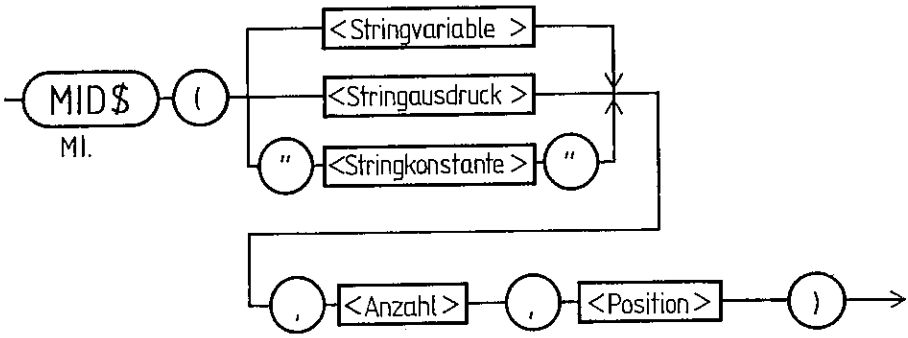


Abbildung E



Siehe auch : LEFT$, RIGHT$

WIRKUNG : MID$ liefert einen Teilstring, der mitten aus einem vorgegebenen String abzulesen ist.

HINWEISE :

<Position> bestimmt, bei welcher Position des Vorgabe-Strings die Zeichenablesung beginnen soll. Die <Position> kann im Bereich 1...80 liegen. Werte, die sich außerhalb davon befinden, haben die Anzeige eines ERROR-Codes zur Folge. Ist die <Position> größer als die Anzahl der im String enthaltenen Zeichen, so wird ein Nullstring geliefert.

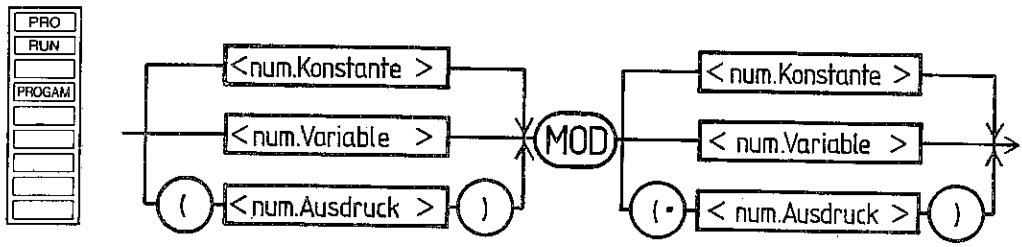
<Anzahl> legt fest, wieviele Zeichen von dem Vorgabe-String zu kopieren sind. Sie kann im Bereich 0...80 liegen. Werte mit Nachkommastellen werden auf die nächste ganze Zahl abgerundet.

BEISPIEL :

```
10:Z$="ABCDEFG"
20:LET Y$=MID$(Z$,3,4)
30:PRINT Y$

>
RUN
CDEF
>
```

MOD



Siehe auch : INT

WIRKUNG : MOD ist ein Operator und liefert von den mit ihm verknüpften Werten den Rest einer ganzzahligen Division.

HINWEIS : Die mit MOD verknüpften Werte sollten ganzzahlig sein. Sie dürfen jedoch durch jede numerische Konstante, numerische Variable oder allgemein durch jeden numerischen Ausdruck vertreten sein.

Ist der Wert einer solchen Konstante, Variable oder Ausdrucks nicht ganzzahlig, so wird er vor Ausführung der Division auf einen ganzzahligen Betrag abgerundet.

BEISPIEL :

```
10:INPUT "DIVIDEND = ";N
20:INPUT "DIVISOR  = ";M
30:R=N MOD M
40:PRINT "REST VON N/M IST: ";R
50:GOTO 10
```

Benutzen Sie die BREAK-Taste, um dieses Programm zu verlassen.

WIRKUNG : MODE selektiert den Anzeige-Modus.

HINWEISE : Das MODE-Kommando kann nur im direkten Betriebsmodus ausgeführt werden, nicht jedoch innerhalb eines Programmes.

MODE 0

Mit MODE oder MODE 0 wird der PC-1600 in einen Anzeige-Modus geschaltet, der alle vier Zeilen des Displays ausnutzt. PRINT-Anweisungen zeigen die Daten in aufeinanderfolgenden Display-Zeilen an. Ist die unterste Display-Zeile beschrieben, wird im PRO-Modus oder bei Benutzung der INPUT-Anweisung der bisherige Display-Inhalt um eine Zeile nach oben geschoben (scrolling), wenn der Platz für eine weitere Datenanzeige nicht ausreicht. Anzeigen, die nicht in eine Zeile von 26 Zeichen passen, setzen sich in der nächsten Display-Zeile fort. In diesem Modus gilt der PC-1600-Zeichensatz (s. Anhang C).

In MODE 0 kann der PC-1600 mit dem speziell auf ihn zugeschnittenen Drucker CE-1600P oder auch anderer PC-1600-Peripherie verwendet werden.

In diesem Modus können keine Programm, die auf dem PC-1500 erstellt worden sind, abgearbeitet werden. Für diese muß MODE 1 eingestellt sein.

MODE 1

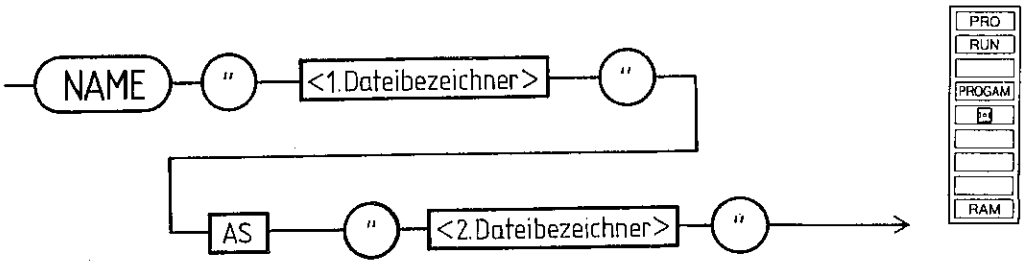
MODE 1 selektiert den PC-1500-kompatiblen Modus. Hierbei wird bei allen Anzeigen nur die unterste Display-Zeile verwendet. Der Zeichensatz wir dem des PC-1500 angepaßt.

Die Länge einer angezeigten Zeile ist auf 26 Zeichen begrenzt. PRINT-Anweisungen, die Daten aufweisen, die nicht mit 26 Zeichen dargestellt werden können, werden auf die ersten 26 Zeichen gekürzt. Aufeinanderfolgende PRINT-Anweisungen löschen den vorhergehenden Inhalt der untersten Display-Zeile.

Ein "Scrollen" des Display-Inhaltes findet nicht statt, mit Ausnahme bei der Erstellung oder dem Editieren von Programmen sowie der Dateneingabe über den INPUT-Befehl.

Im PRO-Modus werden alle vier Zeilen benutzt.

Der Computer kann in diesem Modus zusammen mit den Peripheriegeräten des PC-1500 verwendet werden.



Siehe auch : COPY, FILES

WIRKUNG : NAME benennt eine auf Diskette oder auf RAM-Disk befindliche Datei um.

Der **<1. Dateibezeichner>** bestimmt, welche Datei umzubenennen ist. Er muß aus den drei Angaben Speichermedium (hier: Datenquelle), Dateiname und Extension bestehen.

Der **<2. Dateibezeichner>** bestimmt, unter welchem Namen die Datei fortan auf dem bisherigen Medium geführt werden soll und unter welcher Extension. Im <2. Dateibezeichner> muß also die Angabe des Speichermediums mit der vom <1. Dateibezeichner> identisch sein. Sowohl Extension als auch Dateiname können unverändert übernommen werden, wenn mindestens eine Änderung bei einem dieser beiden Bezeichnungen vorkommt. Bleibt die Extension erhalten, kann ihre Angabe im <2. Dateibezeichner> ebenso weggelassen werden wie die Medium-Angabe.

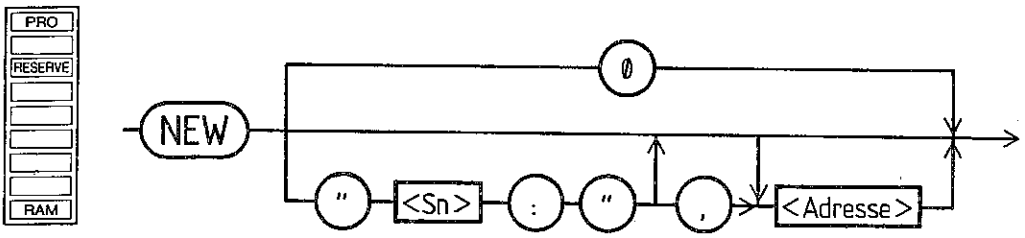
Eine Umbenennung der Datei erfolgt nicht, wenn:

- die Diskette oder RAM-Disk mit einem Schreibschutz versehen ist,

  die Datei mit der P-Option des SET-Kommandos schreibgeschützt ist, oder

- die Datei offen ist.

BEISPIEL :
```
>
NAME "X:ALTEDATEI.XXX" AS "NEUEDATEI.BAS"
```



Siehe auch : DELETE, STATUS, TITLE

WIRKUNG : NEW löscht Programme oder Tastaturbelegungen und reserviert Speicherplatz für Maschinensprache-Programme.

HINWEISE : Im RESERVE-Modus löscht NEW alle Belegungen der Funktionstasten, und zwar aller drei Ebenen.

Gibt man ein Programm in den Speicher ein, ohne das bisher dort befindliche Programm zu löschen, bleiben alle Zeilen mit den Zeilennummern, die in dem neuen Programm nicht vorkommen, weiterhin erhalten. Es entsteht somit ein Programmgemisch, das zu den verschiedensten Ablauffehlern des neuen Programmmes führen kann. Aus diesem Grunde sollte vor Eingabe eines neuen Programmes der Speicher erst einmal vom alten Programm befreit werden. Hierzu dient das Kommando NEW.

PC-1600-Modus (MODE 0)

Der vom Computer adressierbare interne Speicher hat eine Kapazität von 12 KByte = 12288 Bytes. Hiervon stehen dem Anwender maximal 12090 Bytes für eigene Programmzwecke zur Verfügung. Dieser Bereich wird Anwenderbereich genannt und teilt sich in vier Unterbereiche auf:

- Maschinensprache-Bereich
- BASIC-Programm-Bereich
- Freier Bereich
- Variablen-Bereich

Somit ist klar: Je mehr Platz zur Aufnahme von Maschinensprache-Programmen reserviert wird, umso weniger steht einem dann für die Speicherung von BASIC-Programmen zur Verfügung.

NEW
löscht sämtliche BASIC- und Maschinensprache-Programme, die über TITLE spezifiziert worden sind, beläßt jedoch den Bereich zur Aufahme von Maschinensprache-Programmen in seiner bisherigen Größe bestehen.

NEW 0
löscht sämtliche BASIC- und Maschinensprache-Programme, die über TITLE spezifiziert worden sind, und setzt die oberste Bereichsadresse für die Maschinensprache-Programme gleich mit der niedrigsten Adresse (197 = &00C5). Damit steht anschließend kein Platz mehr zur Aufnahme von Maschinensprache-Programmen zur Verfügung.

NEW "Sn:"
löscht alle Programme, die sich in dem durch Sn: bestimmten Speicherbereich befinden:

S0: internes RAM des PC-1600
S1: RAM des im Fach S1 eingesetzten Modules
S2: RAM des im Fach S2 eingesetzten Modules

NEW "Sn:",<Adresse>
löscht entsprechend wie Anweisung NEW "Sn:" und bestimmt mit <Adresse> die oberste Adresse des Maschinensprache-Bereichs, die maximal auf 65535 = &FFFF lauten kann. Die unterste Adresse lautet 197 = &00C5.

PC-1500-Modus (MODE 0)

NEW
löscht alle Programme des Anwenderbereiches und beläßt den reservierten Maschinensprache-Bereich bei der zuvor gesetzten Größe.

NEW 0
löscht alle Programme des Anwenderbereiches und setzt die oberste Adresse des Maschinensprache-Bereichs gleich mit dessen niedrigster Adresse (197 = &00C5). Damit steht kein Speicherplatz für Maschinensprache-Programme zur Verfügung.

NEW <Adresse>
löscht alle Programme des Anwenderbereiches und bestimmt mit <Adresse> die oberste Adresse des Maschinensprache-Bereiches. Die unterste Adresse lautet 197 = &00C5. So ergibt sich bei NEW 1000 beispielsweise folgende Aufteilung:

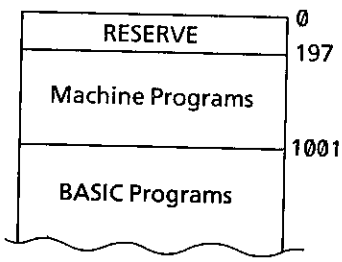
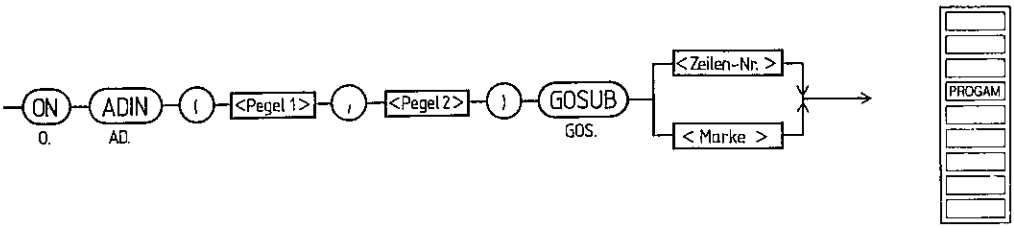


Abbildung F:

BEISPIEL :

```
>
LIST
10:REM DIESES IST EIN KLEINES
20:REM DREIZEILIGES PROGRAMM.
30:END
>

>
NEW
>
LIST
>
```



Siehe auch : ADIN ON/OFF/STOP, AIN, RETI

WIRKUNG : ON ADIN GOSUB verzweigt den Programmablauf in eine Interrupt-Routine, wenn am Analog-Eingang ein vereinbarter Spannungspegel anliegt.

HINWEISE : Die Parameter **<Pegel 1>** und **<Pegel 2>** bestimmen den Pegelbereich der analogen Eingangsspannung. Die erlaubten Werte dieser Parameter liegen im Bereich von 0 bis 255. (Abschnitt 6.6 beschreibt den Zusammenhang zwischen diesen Werten und den Spannungspegeln am Analogeingang.)

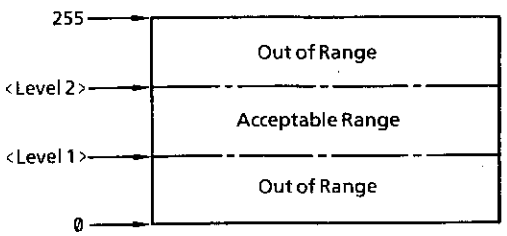


Abbildung 44 : Akzeptierter Pegel-Bereich

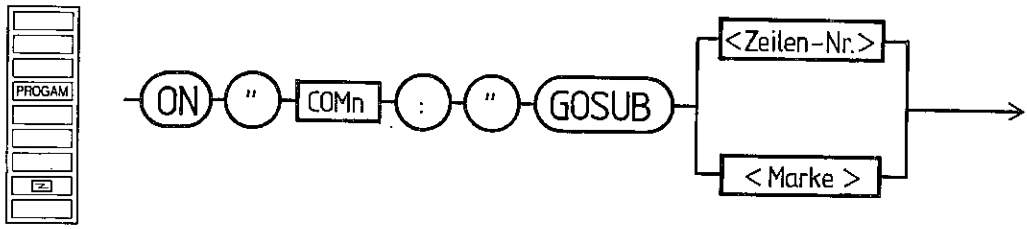
Die Wirkung von ON ADIN GOSUB läßt sich mit der Anweisung ADIN ON/OFF/STOP beeinflussen. Wird keine solche Anweisung nach einem ON ADIN GOSUB angegeben, gilt standardmäßig ADIN STOP.

Die Interrupt-Routine muß mit der RETI-Anweisung abgeschlossen sein.

BEISPIEL :

```
20:ON ADIN(100,165) GOSUB 500
:
490:END
500:REM INTERRUPT-ROUTINE
:
600:RETI
```

ON COMn GOSUB



Siehe auch : COMn ON/OFF/STOP, RETI

WIRKUNG : ON COMn GOSUB verzweigt den Programmablauf in eine Interrupt-Routine, wenn über ein Interface eine Interrupt-Anforderung gestellt wird.

HINWEISE : Der Parameter COMn: ist durch COM1: oder COM2: zu konkretisieren und meint mit:

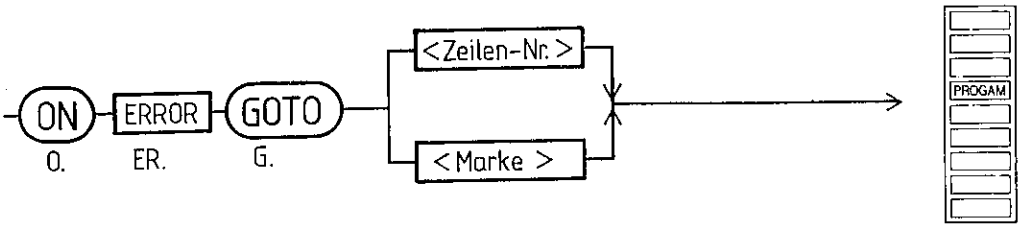
COM1: das RS-232C-Interface
COM2: das SIO-Interface.

Die Wirkung von ON COMn GOSUB läßt sich mit der Anweisung COMn ON/OFF/STOP beeinflussen. Wird keine solche Anweisung gegeben, gilt standardmäßig COMn STOP.

Die Interrupt-Routine muß mit der RETI-Anweisung abgeschlossen sein.

BEISPIEL :

```
20:ON COM1 GOSUB 500
:
:
490:END
500:REM INTERRUPT-ROUTINE
510:REM FUER RS-232C
:
:
600:RETI
```



Siehe auch : ERL, ERN, RESUME

WIRKUNG : ON ERROR GOTO bewirkt, daß beim Auftreten eines Fehlers die Anzeige des zugehörigen ERROR-Codes verhindert und stattdessen in die spezifizierte Fehlerbehandlungs-Routine gesprungen wird.

HINWEISE : Eine Fehlerbehandlungs-Routine kann vom Anwender des Computers geschrieben werden, um mit dieser festzustellen, in welcher Zeile des Programmes ein Fehler aufgetreten und welcher Art dieser ist. Dann kann innerhalb dieser Routine darüber entschieden werden, welche Maßnahmen weiter zu ergreifen sind: ob z.B. durch eine Korrektur der Fehler behoben werden kann, der Benutzer des Programmes zu einer erneuten Dateneingabe aufgefordert wird und dergleichen mehr.

Soll die Fehlerbehandlungs-Routine das Programm weder durch eine STOP- noch eine END-Anweisung beenden, so muß die Rückkehr in das eigentliche Programm über den Befehl RESUME erfolgen.

Tritt in der Fehlerbehandlungs-Routine selbst ein Fehler auf, wird das Programm abgebrochen und ein ERROR-Code angezeigt.

Es gibt keine Begrenzung, wie oft die Anweisung ON ERROR GOTO innerhalb eines Programmes anwendbar ist. Tritt ein Fehler auf, gilt immer die zuletzt ausgeführte ON ERROR GOTO Anweisung.

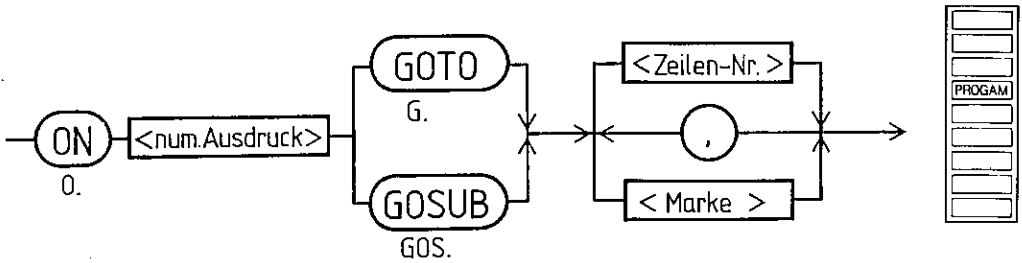
Mit ON ERROR GOTO 0 kann auf die normale Fehlerbehandlung des BASICs zurückgeschaltet werden. Damit wird beim Auftreten eines Fehlers wieder ein entsprechender ERROR-Code angezeigt und der Ablauf des fehlerhaften Programmes abgebrochen.

Die Wirkung einer ON ERROR GOTO Anweisung wird ebenfalls durch die Befehle RUN und END sowie der Betätigung der SHIFT + CL Tasten aufgehoben.

Der Start eines Programmes mit GOTO oder DEF hat keine solche aufhebende Wirkung.

BEISPIEL :

```
5:ON ERROR GOTO 100
10:SAVE "X:DEMO"
20:PRINT "OK"
30:END
:
:
100:IF ERN =160THEN PRINT
    "KEINE DISKETTE IM LAU
    FWERK"
110:PRINT "BITTE DISKETTE
    EINSETZEN"
120:PRINT "UND TASTE J DRU
    CKEN : ";
130:INPUT ""; A$
140:IF A$="J"THEN RESUME
150:STOP
```



Siehe auch : GOSUB..RETURN, GOTO

WIRKUNG : ON...GOSUB und ON...GOTO dienen der bedingten Programmverzweigung in Abhängigkeit des Integer-Wertes eines numerischen Ausdruckes.

HINWEISE : Die als Parameter aufgelisteten **<Zeilennummern>** oder **<Marken>** bestimmen jene Programmzeilen, an die nach folgenden Regeln gesprungen wird:

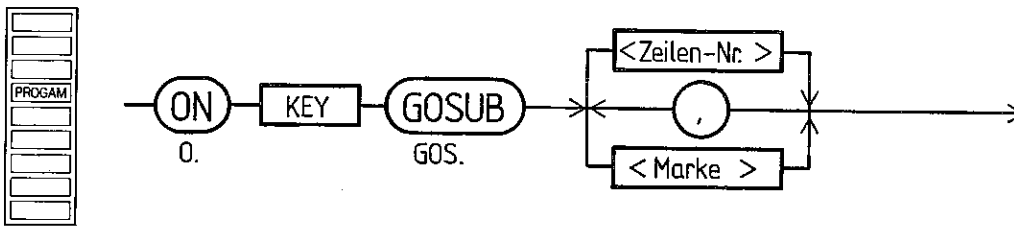
Weist das Verzweigungskriterium **<num. Ausdruck>** den Wert 1 auf, so findet ein Sprung zur ersten Zeile, die in der Parameter-Liste definiert ist, statt. Ergibt der numerische Ausdruck den Wert 2, so wird zur zweiten spezifizierten Zeile gesprungen und so fort. Ist der <num. Ausdruck> mit Nachkommastellen behaftet, wird er auf die nächstniedrigere ganze Zahl (Integer-Wert) abgerundet.

Ist der Wert größer als die Anzahl der in der Liste enthaltenen Elemente, wird das Programm mit der nächsten Programmzeile fortgeführt, was auch für jene Fälle gilt, in denen der Wert kleiner als 1 lautet.

BEISPIEL :

```
10:INPUT "NUMMER (1-3) = ";N
20:ON N GOSUB 100,200,300
:
90:END
100:REM ERSTES UNTERPROGRAMM
:
190:END
200:REM ZWEITES UNTERPROGRAMM
:
290:END
300:REM DRITTES UNTERPROGRAMM
:
390:END
```

ON KEY GOSUB



Siehe auch : KEY ON/OFF/STOP, RETI

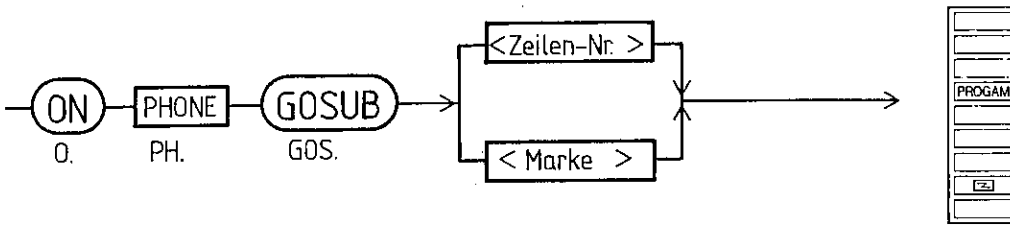
WIRKUNG : ON KEY GOSUB verzweigt in eine von maximal sechs Interrupt-Routinen, wenn dazu eine Aufforderung über eine der sechs Funktionstasten erfolgt.

HINWEISE : Die Wahl der Funktionstaste bestimmt, welche der sechs Interrupt-Routinen abzuarbeiten ist. Wird die erste Funktionstaste (F1) gedrückt, erfolgt ein Sprung zur ersten Routine, die in der Liste der Parameter durch eine **<Zeilen-Nr.>** oder eine **<Marke>** spezifiziert ist. Bei der zweiten Taste (F2) wird zu der an zweiter Stelle genannten Interrupt-Routine verzweigt usw. Enthält die Parameterliste mehr als sechs Elemente, werden die überzähligen Elemente ignoriert. Sind in der Aufzählung weniger als sechs Interrupt-Routinen deklariert, bleiben die entsprechenden Tasten wirkungslos.
Die sechs Funktionstasten lösen nur dann einen Interrupt aus, wenn ein Programm läuft und eine KEY ON Anweisung ausgeführt worden ist. Ohne sie gilt standardmäßig KEY STOP, was die Annahme des Interrupts verhindert, aber zwischenspeichert.

Jede Interrupt-Routine muß mit der Anweisung RETI abgeschlossen sein, damit eine Rückkehr in den normalen Programmablauf möglich ist.
RUN und END löschen sämtliche Zuordnungen von Interrupt-Routinen und Funktionstasten, so daß diese Tasten wieder normal belegt sind.

BEISPIEL :

```
10:ON KEY GOSUB 100,200
20:KEY (1)ON :KEY (2)ON
30:PAUSE "*";:GOTO 30
100:FOR I=1TO 6:PAUSE "1";
    :NEXT I:RETI
200:FOR J=1TO 6:PAUSE "2";
    :NEXT J:RETI
```



Siehe auch : PHONE ON/OFF/STOP, RETI

WIRKUNG : Mit ON PHONE GOSUB wird der Programmablauf in eine Interrupt-Routine verzweigt, wenn über das RS-232C-Interface eine entsprechende Anforderung an Pin 9 (CI-Signal) anliegt.

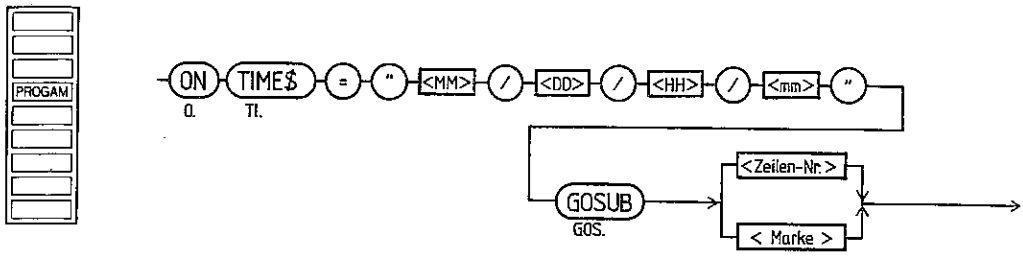
HINWEISE : Die Interrupt-Routine muß mit der Anweisung RETI abgeschlossen sein.

Die Wirkung von ON PHONE GOSUB läßt sich mit der Anweisung PHONE ON/OFF/STOP beeinflussen. Ohne eine solche Anweisung wird standardmäßig so verfahren, als sei PHONE STOP angegeben worden.

BEISPIEL :

```
10:ON PHONE GOSUB 500
20:PHONE ON
:
190:PHONE OFF
:
:
250:'PHONE INTERRUPT
:
:
290:RETI
```

ON TIME$ GOSUB



Siehe auch : RETI, TIME$, TIME$ ON/OFF/STOP

WIRKUNG : ON TIME$ GOSUB verzweigt den Programmablauf in eine Interrupt-Routine, wenn ein vereinbarter Zeitpunkt erreicht ist.

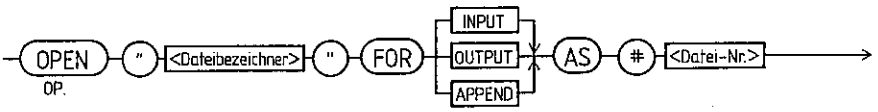
HINWEISE : Die Spezifikation des gewünschten Zeitpunktes erfolgt im selben Format wie bei der Stellung der Uhr über TIME$.

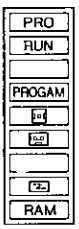
Ist die betreffende Zeit erreicht, wird die Interrupt-Routine, die hinter dem Befehlswort GOSUB durch eine <Zeilen-Nr.> oder eine <Marke> bestimmt ist, aufgerufen.

Sie muß, damit eine Rückkehr in das normale Programm möglich ist, mit der RETI-Anweisung abgeschlossen sein.

Die Anweisung ON TIME$ GOSUB läßt sich mittels TIME$ ON/OFF/STOP in ihrer Wirkung beeinflussen. Standardmäßig wird die Einstellung TIME$ STOP angenommen.

BEISPIEL : 10:ON TIME$="05/12/15/30/" GOSUB 500





WIRKUNG : OPEN öffnet eine Datei und erlaubt somit, auf diese zum Zwecke des Lesens, Schreibens oder Anhängens von Daten zugreifen zu können.

HINWEISE : Je nach gewünschter Art des Zugriffes ist dem OPEN-Befehl ein entsprechendes Attribut (INPUT, OUTPUT oder APPEND) beizufügen und die Datei in diesem Modus zu öffnen. Auf die Datei kann dann nur zu genau diesem Zwecke zugegriffen werden. Bevor eine Datei zu einem anderen Zweck geöffnet werden kann, muß sie zuvor mit CLOSE geschlossen werden.

INPUT erlaubt eine Dateiöffnung, um von der Datei Datensätze sequentiell mittels INPUT# lesen zu können.

OUTPUT gestattet eine Öffnung der Datei, um in diese sequentiell Datensätze mit PRINT# hineinschreiben zu können. In diesem Modus können keine Datensätze an die Datei angefügt werden.

APPEND öffnet die Datei in der Weise, daß an ihren bisherigen Inhalt mit Hilfe von PRINT# weitere Datensätze anfügbar sind.

Der Parameter **<Dateibezeichner>** bestimmt, auf welchem Medium die Datei zu finden ist und unter welchem Namen und welcher Extension.

Mit dem Parameter **<Datei-Nr.>** wird der Datei eine Nummer zugewiesen, unter der auf sie mit den weiteren Dateibefehlen zugegriffen werden kann. Diese Nummer darf zwischen 1 und dem über MAXFILES bestimmten Maximum liegen.

Die Öffnung einer Datei bleibt erfolglos, wenn:

- der OUTPUT-Modus gewünscht ist, aber die Datei mit einem Schreibschutz versehen ist,

- ein serielles Interface über COM1: bzw. COM2: im APPEND-Modus spezifiziert wird.

BEISPIEL :

```
5:MAXFILES=1
10:OPEN "X:DATA"FOR OUTPUT AS #1
20:FOR J=1TO 5
30:PRINT #1,J
40:NEXT J
50:CLOSE #1
60:OPEN "X:DATA"FOR INPUT AS #1
70:IF EOF(1)THEN 110
80:INPUT #1,J
90:PRINT J
100:GOTO 70
110:REM DATEIENDE ERREICHT
120:CLOSE #1
130:END
```

Zeile 10 eröffnet eine neue Disketten-Datei, so daß in diese Daten hineingeschrieben werden können.

Zeile 30 schreibt die Werte 1 bis 5 direkt aufeinanderfolgend in diese Datei hinein.

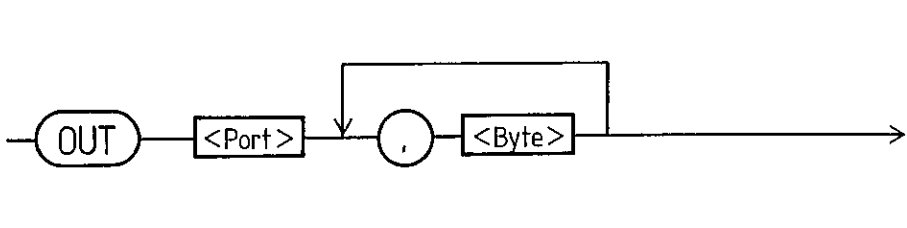
Zeile 50 schließt die Datei, damit sie gleich zu einem anderen Zweck geöffnet werden kann.

Zeile 60 öffnet die Datei zum Lesen der Daten.

Zeile 70 prüft, ob das Dateiende erreicht ist und verzweigt im zu bejahenden Falle zur Zeile 110.

Zeile 80 liest die Daten.

Zeile 90 zeigt die Daten an.



Siehe auch : INP

WIRKUNG : OUT gibt ein Byte über den gewünschten Port des Z80-kompatiblen Mikroprozessors aus.

HINWEISE : **<Port>** ist eine Adresse (16-Bit-Wert) im Werte-Bereich von 0...65535 (&0...&FFFF), die den gewünschten Port selektiert.

<Byte> gibt das an den Port zu liefernde Byte an. Werden mehrere Bytes als Parameter aufgelistet, so wird jedes folgende Byte an die nächste Port-Adresse abgegeben. Aufeinanderfolgende Bytes werden also an aufeinanderfolgende Ports übergeben.

BEISPIEL : OUT 80,187

Diese Anweisung sendet den Wert 187 = &BB an den Port 80 (=&50).

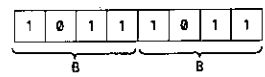
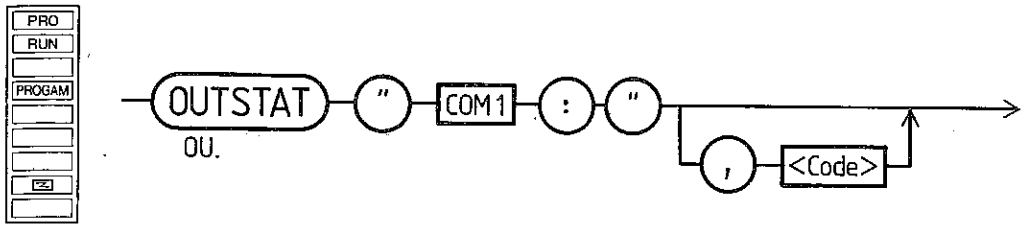


Abbildung G

OUTSTAT



Siehe auch : INSTAT

WIRKUNG : OUTSTAT bestimmt den Zustand der Steuersignale RTS und DTR des RS-232C-Interface.

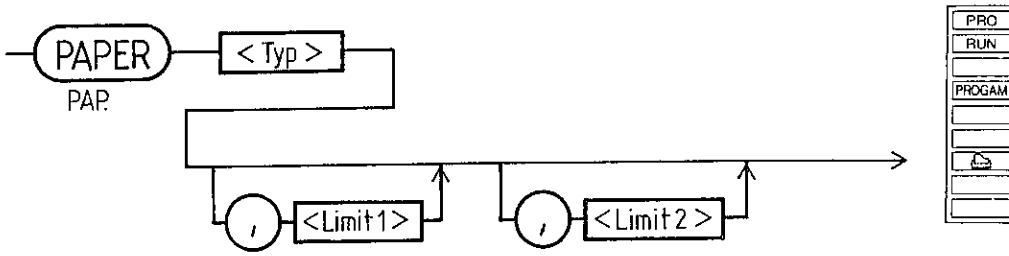
HINWEISE : Die Zustände der Signale RTS (request to send) und DTR (data terminal ready) werden durch den Parameter <Code> wie folgt bestimmt:

<Code>	RTS	DTR
0	high	high
1	high	low
2	low	high
3	low	low

Fehlt dieser Parameter, bleiben beide Signale während der Ausführung eines Interface-Befehles "high". Davor oder danach werden sie auf "low" gesetzt. Ist der Empfangspuffer gefüllt, so geht RTS auf "low", um damit den Datentransfer zu stoppen und den Puffer entleeren zu können.

BEISPIEL :
```
>
OUTSTAT "COM1:",2
```

Diese Anweisung setzt das RTS-Signal des Ports auf "low" und das DTR-Signal auf "high".



WIRKUNG : PAPER teilt dem Computer mit, welches Format das in den Drucker eingespannte Papier aufweist, und bestimmt, in welchem vertikalen Bereich gedruckt werden darf.

HINWEISE : <Typ> beschreibt das Papierformat wie folgt:

C = Einzelblattpapier und
R = Endlospapier (Papierrolle).

Die beiden anderen Parameter sind optional und legen den zulässigen Druckbereich in Y-Richtung in Schritten von 0.2mm fest. Fehlen diese Parameter, werden standardmäßige Werte angenommen.

Die mit PAPER definierten Grenzen beziehen sich auf die momentane Stiftposition und werden mit Ausführung dieses Befehles unmittelbar wirksam. Mit Betätigung der am Drucker befindlichen Taste [↑], dem Einschalten des Druckers sowie der Ausführung der Befehle TEXT oder GRAPH sind die Begrenzungen auf die dann gerade vorliegende Stiftposition zu beziehen.

<Limit 1> bestimmt die obere Druckgrenze, also um wieviele Einheiten zu je 0.2mm das Papier rückwärts transportiert werden kann. Hier sind Einheiten zwischen 30 und 2047 erlaubt. Die Standardwerte lauten:

30 (= 6.0mm) für Einzelblatt- und
999 (=199.8mm) für Endlospapier.

<Limit 2> legt die untere Druckgrenze fest, d.h. um wieviele Einheiten zu je 0.2mm das Papier vorwärts bewegbar ist. Auch hier sind für den Parameter Werte von 30 bis 2047 zulässig.

Als Standardwerte für <Limit 2> gelten folgende:

1354 (=272.8mm) für Einzelblatt- und
999 (=199.8mm) für Endlospapier.

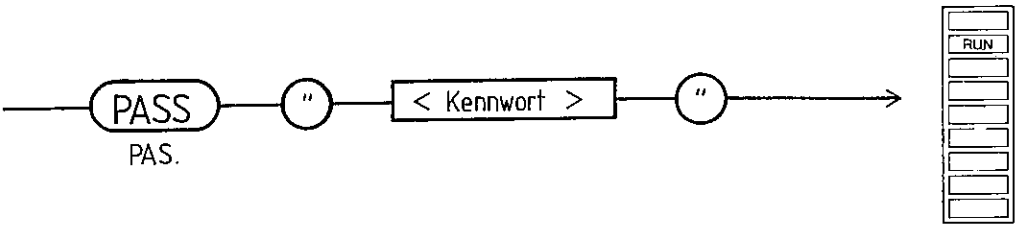
Bei der Ausführung der Befehle GRAPH und TEXT werden die Einstellungen auf die Standardwerte von <Limit 1> und <Limit 2> zurückgesetzt. Im Text-Modus gilt dabei für <Limit 2> der Wert für Endlospapier.

Ein Versuch, im Grafik-Modus den Druckstift über die mit PAPER festgelegten Grenzen hinauszubewegen, spiegelt die Projektion des verhinderten Druckweges an diesen Bereichsgrenzen.

BEISPIEL :

```
>
PAPER C
```

Dieses Kommando bereitet den Drucker auf den Gebrauch von Einzelblatt-Papier im DIN-A4-Format vor. Die Druckgrenzen werden auf die Standardwerte eingestellt.



Siehe auch : CLOAD, CSAVE, SET

WIRKUNG : Das PASS-Kommando erlaubt es, ein Programm durch die Vergabe eines Kennwortes (password) gegen den unerlaubten Zugriff fremder Personen zu schützen. Ein solches **<Kennwort>** besteht aus bis zu acht beliebig kombinierten Zeichen, die wie eine Stringkonstante in Anführungsstriche einzuklammern sind. Das Anführungszeichen " kann somit nicht innerhalb des Kennwortes verwendet werden.

Nachdem ein Kennwort gesetzt worden ist, kann der Computer nicht mehr in den PRO- oder den RESERVE-Modus versetzt werden. Die nachstehend genannten Befehle bleiben dann ebenso wirkungslos wie die Tasten [↓] und [↑]:

LIST, LLIST, CSAVE, SAVE, CLOAD, LOAD, NEW, TITLE, MERGE und CHAIN.

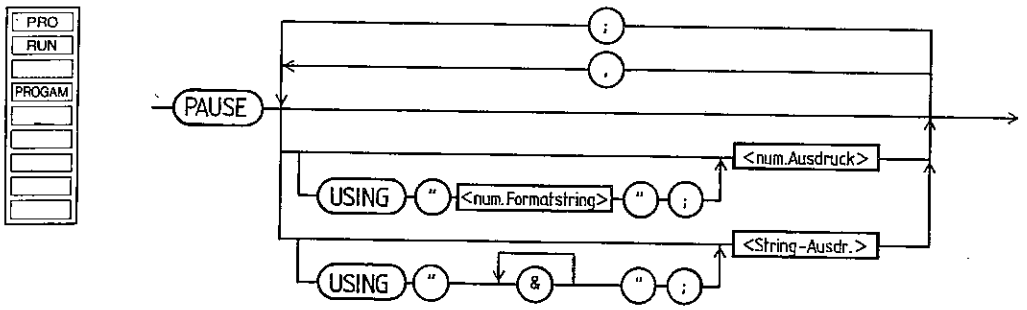
Damit kann das Programm weder aufgelistet, gespeichert noch verändert werden. Ebenso ist ein Überschreiben durch Laden eines anderen Programmes verhindert. Befinden sich mehrere Programme im Speicher, so gilt diese Schutzwirkung für alle diese Programme. Der einzige Weg, um den Schutz aufzuheben, ist, das PASS-Kommando mit dem betreffenden Kennwort nochmalig einzugeben.

Das PASS-Kommando ist nur dann anwendbar, wenn sich im Programmspeicher auch tatsächlich ein Programm befindet.

BEISPIEL : PASS "GEHEIM"

Dieses Kommando schützt alle im Speicher vorhandenen Programme durch das Kennwort "GEHEIM".

PAUSE



Siehe auch : PRINT, WAIT

WIRKUNG : PAUSE zeigt die als Parameter angegebenen Daten auf dem Display an und wartet mit der weiteren Programmausführung für eine definierte Zeit von 0.85 Sekunden.

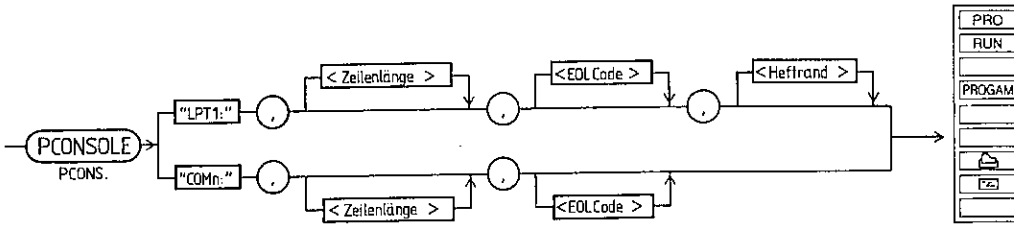
Damit wirkt PAUSE wie eine PRINT-Anweisung, der ein entsprechendes WAIT-Kommando vorausgegangen ist. Die Anwendung von PAUSE erweist sich als besonders nützlich, wenn der Computer im PC-1500 kompatiblen MODE 0 betrieben wird. Hier würden sonst alle Ausgaben von PRINT-Anweisungen sofort aus der einen Display-Zeile herauswandern und damit unsichtbar bleiben.

In MODE 1 kann die Anzahl der als Parameter aufgeführten Ausdrücke maximal zwei betragen, wenn als Trennzeichen ein Komma verwendet wird.

Da PAUSE und PRINT sich nur in ihrer Anzeigedauer voneinander unterscheiden, verhalten sie sich auch bezüglich der USING-Anweisung gleich. Näheres kann hierzu den Erklärungen von USING entnommen werden.

BEISPIEL :

```
10:PAUSE "LIES SCHNELL DIESEN SATZ !"
20:CLS
30:END
```



WIRKUNG : Mit PCONSOLE kann die Zeilenlänge und der Code für das Zeilenende sowohl für den Drucker als auch die seriellen Interfaces bestimmt werden.

HINWEISE : **Einstellung für den Drucker**

Lautet der erste Parameter **"LPT1:"**, so bezieht sich die mit PCONSOLE vorzunehmende Einstellung auf den derzeitig aktivierten Drucker.

Die **<Zeilenlänge>** bestimmt, wieviele Zeichen in einer Druckzeile enthalten sein können. Der Wert darf zwischen 16 und 255 liegen. Gibt man jedoch den Wert 0 an, wird damit die Zeilenlänge auf unendlich gesetzt.

Der **<EOL-Code>** oder auch Zeilenend-Code (end of line code) genannt, bestimmt, welche Zeichen zur Kennung des Zeilenendes zum Drucker gesendet werden, wenn die Befehle LLIST, LPRINT oder LFILES zur Ausführung kommen. Es gilt:

<EOL-Code>	Zeichen	Hex-Code(s)
0	<CR>	&0D
1	<LF>	&0A
2	<CR> + <LF>	&0D + &0A

Andere Angaben führen zur Anzeige eines ERROR-Codes.

Der **<Heftrand>** legt fest, wieviele Leerzeichen der eigentlichen Ausgabe vorausgeschickt werden sollen, damit ein Rand zum Abheften des Papieres verbleibt. Für diesen Wert muß gelten:

<Heftrand> ≤ <Zeilenlänge> - 4

Die standardmäßig vorausgesetzten Einstellungen lauten:

- unendliche Zeilenlänge
- <EOL-Code> = 0
- <Heftrand> = 0

Falls die mit LPRINT ausgedruckten Zeichen den Stift weniger als 4 mm bewegen (z.B. CSIZE 1), kehrt dieser nicht zum linken Rand zurück, wenn ein Zeilenvorschub stattfindet.

Einstellungen für die seriellen Ports

Wird als erster Parameter **"COMn:"** angegeben, so beziehen sich die dann folgenden Parameter auf eine der beiden seriellen Schnittstellen gemäß:

COM1:	RS-232C-Interface
COM2:	SIO-Interface
COM:	über SETDEV selektiertes Interface

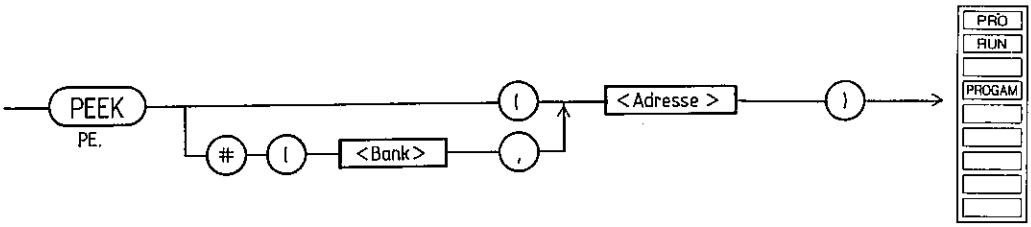
Die **<Zeilenlänge>** bestimmt in diesem Falle, wieviele Zeichen über den Port ausgesendet werden können, bevor eine Trennung der einzelnen Zeilen mittels <EOL-Code> erfolgt. Auch hier gelten die schon für den Drucker genannten Parameterwerte.

<EOL-Code> bestimmt was auf dem Port auszugeben ist, um das Zeilenende zu signalisieren. Hierbei sind dieselben Einstellungen wie für den Drucker möglich. Der Code bestimmt darüberhinaus, welche Zeichenfolge als einfacher "carriage return" zu interpretieren ist, wenn Daten mit einer INPUT-Anweisung über den Port eingelesen werden.

Alle über PCONSOLE vorgenommenen Einstellungen bleiben solange wirksam, bis sie durch ein neues PCONSOLE-Kommando überschrieben werden.

Wird die Zeichengröße mit CZISE variiert, ändert sich die über PCONSOLE definierte Zeilenlänge in entsprechender Proportion. Die Einstellung des Heftrandes wird davon jedoch nicht berührt.

BEISPIEL : PCONSOLE "LPT1:",42,2,3



Siehe auch : POKE, XPEEK, XPOKE

WIRKUNG : PEEK liefert den Inhalt einer spezifizierten Speicheradresse.

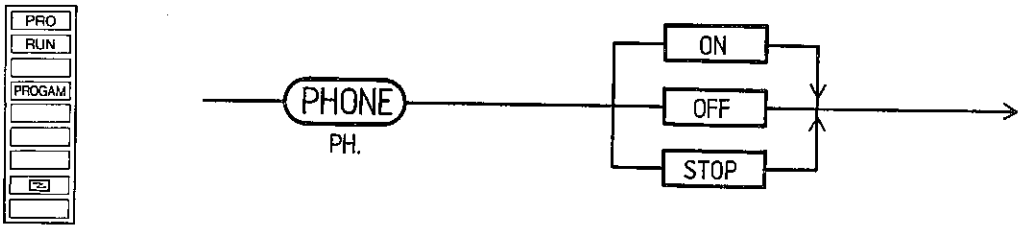
HINWEISE : **<Adresse>** bestimmt die Adresse im Bereich von &0 bis &FFFF (0 bis 65535) bezüglich der jeweils gültigen Speicherbank.

<Bank> beschreibt, auf welche Speicherbank sich die Angabe der Adresse bezieht. Es sind hier die Werte von 0 bis 7 angebbar. Fehlt der Parameter <Bank>, wird hierfür standardmäßig der Wert 0 angenommen. Näheres zur Aufteilung des Speichers in "memory banks" geht aus Anhang D hervor.

BEISPIEL :
```
>
PEEK (0,100)
```

Dieser Befehl liefert den Inhalt der Speicherzelle 100 (=&64) der Speicherbank 0.

PHONE ON/OFF/STOP



Siehe auch : ON PHONE GOSUB

WIRKUNG : Erlaubt oder verbietet dem Computer die Annahme von Interrupt-Anforderungen, die von einem Modem über das RS-232C-Interface gestellt werden.

HINWEISE : **PHONE ON**

Mit dem Befehl PHONE ON wir die Annahme solcher Interrupt-Anforderungen zugelassen. Tritt eine Anforderung über das RS-232C-Interface auf kann mit der Anweisung ON PHONE GOSUB in die entsprechende Interrupt-Routine verzweigt werden.

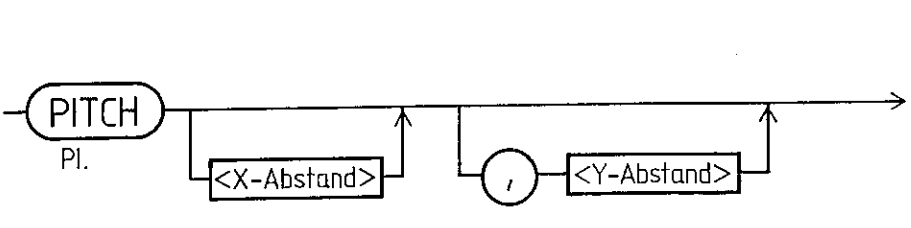
PHONE OFF

Über den Befehl PHONE OFF wird die Annahme der genannten Interrupt-Anforderungen verhindert.

PHONE STOP

Durch den Befehl PHONE STOP wird die Annahme der über das RS-232C-Interface gestellten Interrupt-Anforderungen ebenfalls verhindert. Der Unterschied zu PHON OFF besteht jedoch darin, daß die jeweils letzte Anforderung zwischengespeichert wird. Bei der nächsten Ausführung eines PHONE ON kommt sie dann unmittelbar zur Geltung. STOP ist die standardmäßig angenommene Einstellung nach Inbetriebnahme des Computers.

BEISPIEL :
```
10:ON PHONE GOSUB 250
20:PHONE ON
:
90:PHONE OFF
```



Siehe auch : CSIZE

WIRKUNG : PITCH bestimmt den im Text-Modus vom Drucker einzuhaltenden Zeichen- und Zeilenabstand.

HINWEISE : Der Parameter <X-Abstand> legt den horizontalen Abstand zweier Zeichen fest und <Y-abstand> den vertikalen Abstand, d.h. den Zeilenabstand.

<X-Abstand>	Abstand zweier Zeichen zueinander
Standard	6 * CSIZE <Größe> * 0.2mm
4-255	<X-Abstand> * 0.2mm

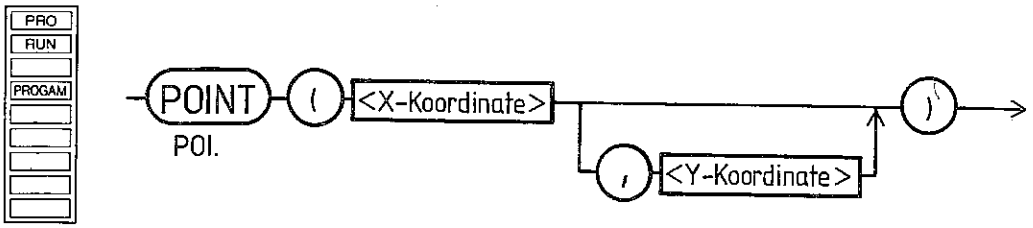
<Y-Abstand>	Abstand zweier Zeilen zueinander
Standard	12 * CSIZE <Größe> * 0.2mm
4-255	<Y-Abstand> * 0.2mm

Fehlt die Angabe eines dieser Parameters, gilt für ihn die Standardeinstellung.

Bei Ausführung der Befehle TEXT, GRAPH, CSIZE, LLIST, TEST und PCONSOLE werden die standardmäßigen Werte für PITCH angenommen. PITCH ist auch im Grafik-Modus verwendbar. Es wird dann allerdings der Parameter <Y-Abstand> ignoriert.

BEISPIEL :
```
10:PITCH 30,40
20:LPRINT "ERSTE  ZEILE"
30:LPRINT "ZWEITE ZEILE"
40:LPRINT "DRITTE ZEILE"
50:END
```

Probieren Sie dieses Programm auch mit anderen PITCH-Parameterwerten aus, um ihren Einfluß auf das Druckverhalten kennenzulernen.



Siehe auch : GPRINT, PRESET, PSET

WIRKUNG : POINT liefert eine Information über den Zustand des spezifizierten Display-Punktes.

HINWEISE : **POINT (X,Y)**

Im Format POINT (X,Y) bestimmen die Koordinaten X und Y den Display-Punkt. Ist der Punkt dunkel, also gesetzt, so liefert POINT (X,Y) den Wert 1, im anderen Falle den Wert 0. Die Koordinaten können in folgenden Bereichen liegen:

<X-Koordinate>	0	= Anfang der Display-Zeile
	155	= Ende der Display-Zeile
<Y-Koordinate>	0	= oberste Punktreihe
	31	= unterste Punktreihe

Obwohl die <X-Koordinate> und die <Y-Koordinate> in diesen Bereichen liegen sollten, um einen realen Display-Punkt zu adressieren, können sie dennoch jeweils im Bereich -32768...32767 angegeben werden. Das gelieferte Ergebnis lautet in diesen Fällen 0. Dieses gilt auch für das nachfolgend beschriebene Format.

POINT (X)

Im Format POINT (X) wird die momentane Display-Zeile in fortlaufende Spalten mit der Breite eines Punktes und der Höhe von acht Punkten aufgeteilt aufgefaßt. X bestimmt die Spalte, von der der Zustand der acht Punkte, als Bitmuster gewertet, geliefert werden soll. Die Zählung beginnt bei Null.

POINT (25) liefert somit den Wert für das in der 26. Punktspalte stehende Bitmuster an. Lautet er beispielsweise 98 (binär: 01100010), so liegt damit das folgende Punktmuster vor:

```
| |  0     niederwertigstes  Bit
|X|  1
| |  0
| |  0
| |  0
|X|  1
|X|  1
| |  0     höchstwertigstes  Bit
```

Unter den Beschreibungen zu GPRINT finden Sie weitere Details zur Bitmuster-Darstellung.

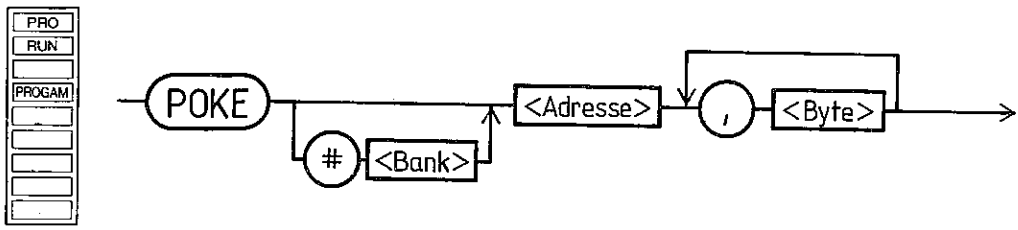
BEISPIEL :

```
10:CURSOR 0,0
20:P=POINT (3):Q=POINT (4)
30:P$=HEX$ P: Q$=HEX$ Q
40:CURSOR 0,1
50:PRINT P;Q
60:PRINT " "+P$+" "+Q$
```

Starten Sie dieses Programm mit dem Befehlswort RUN, so liefert es die Werte der Bitmuster, die die letzten beiden Punktspalten des Buchstabens R bilden. Es ergibt sich dann folgende Anzeige:

```
>
RUN
 41  70
 29  46
>
```

Die erste Zahlenreihe gibt die Werte dezimal, die darunterstehende Zeile hexadezimal an.



Siehe auch : PEEK, XPEEK, XPOKE

WIRKUNG : Mit POKE besteht ein direkter Zugriff auf die Speicherzellen des Computers. Es können hiermit Daten, die in Form von Bytes vorliegen, gezielt in die spezifizierten RAM-Adressen geschrieben werden.

HINWEISE : **<Adresse>** bestimmt, in welche Speicherzelle das (erste) angegebene Byte zu schreiben ist. Die im Bereich von 0..65535 bzw. &0...&FFFF liegende Adresse bezieht sich dabei auf die derzeit gültige oder die über <Bank> spezifizierte Speicherbank.

<Bank> ist eine ganze Zahl zwischen 0 und 7. Sie legt die gewünschte Speicherbank fest, auf die sich die Adresse beziehen soll. Fehlt dieser Parameter, so gilt diejenige Bank, in der sich der Datei-Kopf des gerade laufenden Programmes befindet. Über die Unterteilung des Speichers in Bänke gibt Anhang D Auskunft.

<Byte> bestimmt einen 8-Bit-Wert im Bereich von 0 bis 255 (bzw.: &0..$FF), der in die durch <Adresse> und <Bank> genau festgelegte Speicherzelle geschrieben werden soll.

Gibt man mehrere Bytes an, die dann mit Kommas voneinander getrennt sein müssen, so werden sie der Reihe nach in aufeinanderfolgende Adressen geschrieben. Der Parameter <Adresse> wirkt dabei als Anfangsadresse.

Reicht der zur Verfügung stehende Speicherplatz nicht aus, um alle aufgelisteten Bytes unterzubringen, so wird ein ERROR-Code angezeigt.

BEISPIEL : Das nachfolgende Beispiel, das nicht nur zur Anwendung des POKE-Befehles, sondern auch der Befehle CALL und PEEK dienen möge, zeigt, wie man ein Maschinensprache-Programm in Z80-Code eingibt, es überprüft und startet.

Als Programm soll eine kleine Routine dienen, die den Wert einer Variablen übernimmt, ihn inkrementiert und an die Variable zurückgibt.

Diese Routine lautet folgendermaßen:

Adresse	Code	Mnemonik
&FF00	&13	INC DE
&FF01	&37	SCF
&FF02	&C9	RET

Sie kann mit nachstehender Anweisung in den Speicher gePOKEd werden:

```
POKE &FF00,&13,&B7,&37,&C9
```

Überprüfen Sie anschließend durch Stichproben, ob diese Werte auch tatsächlich in den Speicher übernommen worden sind. Zum Beispiel:

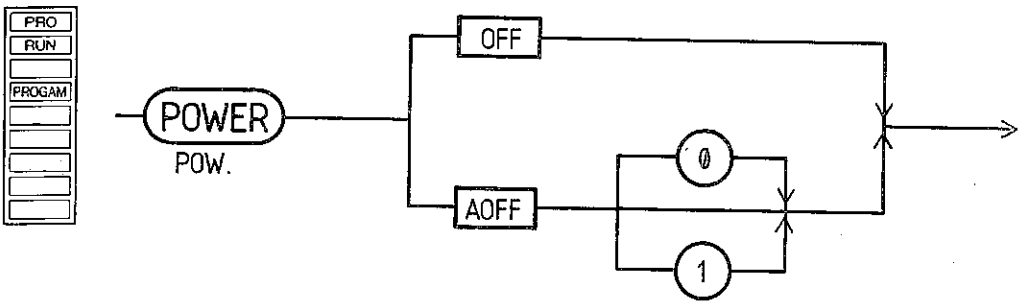
```
>
HEX$ PEEK &FF02
C9
>
```

Ist alles in Ordnung, kann die Routine gestartet werden. Als Übergabevariable soll in diesem Beispiel die Variable X dienen, die zuvor mit einem Testwert zu belegen ist:

```
>
CLEAR
X=77
CALL &FF00,X
```

Fragen Sie dann die Variable ab, ob wirklich ihr Wert um den Betrag 1 inkrementiert worden ist:

```
>
X
78
>
```



Siehe auch : ALARM$, ARUN, WAKE$

WIRKUNG : POWER schaltet die "auto power off"-Funktion ein bzw. aus oder ermöglicht die programmgesteuerte Ausschaltung des Computers.

HINWEISE :

POWER OFF schaltet den Computer unverzüglich aus, wenn dieses Kommando in einem laufenden Programm aktiviert wird.

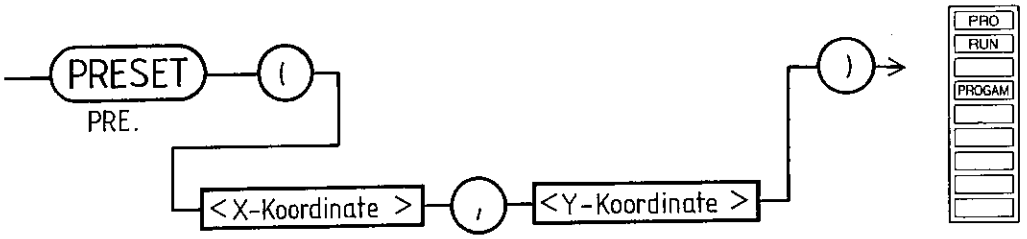
Im RUN- oder PRO-Modus wird dieses Kommando auch bei Eingabe über die Tastatur wirksam.

POWER AOFF
POWER AOFF 0 Mit diesen beiden Kommandos kann die "auto power off"-Funktion des PC-1600 aktiviert werden. Diese sorgt dafür, daß sich der Computer nach ca. 10 Minuten selbsttätig abschaltet, sofern kein Programm läuft und über die Tastatur innerhalb dieser Zeitspanne keine Eingaben getätigt wurden.

POWER AOFF 1 schaltet diese "auto power off"-Funktion ab.

Werden andere Werte als 0 oder 1 als Parameter dem Befehl POWER AOFF nachgestellt, so bleibt das Kommando wirkungslos.

Nach einem Total-Reset ist die "auto power off"-Funktion aktiv. Ebenso läßt sich der Computer dann über ein Modem oder zu einer mit WAKE$ bestimmten Zeit automatisch einschalten.



Siehe auch : PSET, LINE

WIRKUNG : PRESET löscht den Display-Punkt, der durch die angegebenen Koordinaten bestimmt ist.

HINWEISE : Die Koordinaten X und Y können im Bereich von -32768 bis 32767 liegen, ohne daß ein ERROR-Code generiert wird. Es sollten jedoch die folgenden Bereiche beachtet werden, wenn ein tatsächlich existierender Display-Punkt anzusprechen ist:

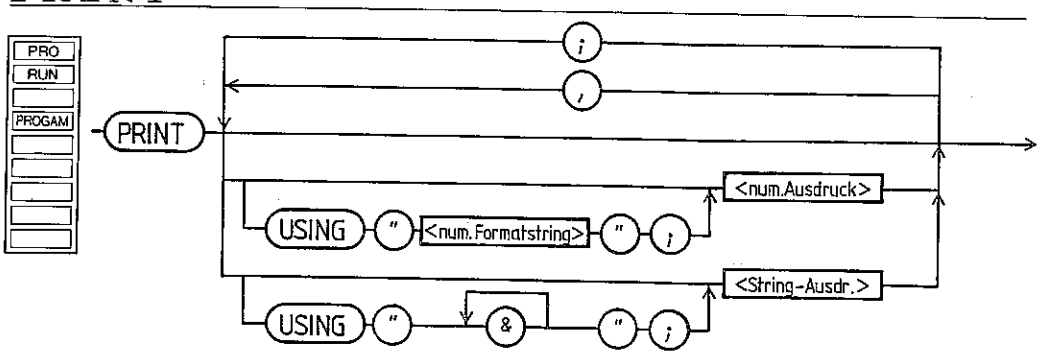
<X-Koordinate>	:	0...155
<Y-Koordinate>	:	0....31

Der gelöschte Punkt erscheint in der Anzeige hell.

BEISPIEL :

```
10:LINE (0,0)-(155,35),,,BF
20:FOR I=1TO 2000
30:X=RND(155): Y=RND(35)
40:PRESET(X,Y)
50:NEXT I
```

PRINT



Siehe auch : MODE, USING, WAIT

ERKLÄRUNG : PRINT gibt Daten auf dem Display aus.

HINWEISE : Wie die Daten dabei auf dem Display positioniert werden, hängt von folgenden Faktoren ab:

a) ob es sich um einzelne Daten handelt oder um eine Auflistung davon,

b) ob für die Auflistung als Trennungszeichen ein Komma (,) oder ein Semikolon (;) verwendet wird, und

c) ob sie numerischen Typs sind oder einen String darstellen.

Bevor Sie die nachfolgenden Erklärungen schrittweise durch einzelne Experimente nachvollziehen, sollten Sie wissen, daß das BASIC jede Zeile des Displays in zwei gleichgroße Zonen unterteilt. Jede Zone besteht damit aus 13 Spalten. Diese Zonen-Einteilung wird immer dann von der PRINT-anweisung berücksichtigt, wenn einzelne Daten vorliegen oder in der Auflistung ein Komma enthalten ist. Mit einem Semikolon voneinander getrennte Daten werden auf dem Display unmittelbar hintereinandergefügt angezeigt.

Wird die Zoneneinteilung berücksichtigt, weil in der PRINT-Anweisung nur ein Parameter beigefügt ist oder aber in der Auflistung ein Komma vorliegt, so erfolgt die Anzeige innerhalb einer solchen Zone :

bei numerischen Daten rechtsbündig,
bei String-Daten jedoch linksbündig.

Numerische Einzel-Daten werden in der rechten, einzelne String-Daten in der linken Zone dargestellt.

Beispiele zonenorientierter Ausgaben:

```
PRINT "A"
A

PRINT 2
                          2

PRINT "A",2
A                         2

PRINT 2,"A"
              2A

PRINT 1,2,3
              1           2
              3

PRINT "A","B",3
A              B
              3
```

Im Anzeige-Modus MODE 1 kann die Liste nur zwei Daten enthalten, wenn das Komma zur Trennung verwendet wird.

In MODE 1

Anders als bei der LPRINT-Anweisung kann die Option TAB nicht in Verbindung mit der Anweisung PRINT stehen. Die Daten lassen sich aber durch die Anwendung der Option USING und der Anweisung CURSOR beliebig positionieren und formatiert darstellen.

Wird eine PRINT-Anweisung durch ein Semikolon abgeschlossen, so setzt sich die nächste Ausgabe in derselben Zeile fort. Schließt dagegen die PRINT-Anweisung mit einem Komma ab, erfolgt die nächste Ausgabe in der nächsten Zone. Beendet weder ein Semikolon noch ein Komma die PRINT-Anweisung, so setzt sich die nächste Ausgabe in der folgenden Zeile fort:

```
10:PRINT 2;
20:PRINT "A"

>
RUN
 2A
>

10:PRINT 2,
20:PRINT "A"

>
RUN
              2A
>

10:PRINT 2
20:PRINT "A"

>
RUN
                              2
A
>
```

Die als Parameter aufgeführten Daten brauchen nicht, wie es in den bisherigen Beispielen der Fall gewesen ist, aus Konstanten zu bestehen, sondern können beliebige numerische Ausdrücke oder Stringausdrücke sein. Beispiel:

```
10:N$="SHARP "
20:PRINT N$+" PC";-(2+INT(1/EXP(-7.377)))

>
RUN
SHARP PC-1600
>
```

Eine PRINT-Anweisung ohne jegliche Parameterangabe bewirkt die Ausgabe einer Leerzeile, was praktisch mit einem Zeilenvorschub beim Drucker verglichen werden kann.

Ein PRINT-Befehl kann mit einer oder mehreren USING-Anweisungen versehen sein, so daß sich die Daten formatiert anzeigen lassen. Das jeweilige Format wird durch einen <Format-String>, der der USING-Anweisung als Parameter beizugeben ist, bestimmt.

Der <Format-String> besteht aus einer Reihe von speziellen Zeichen, die in Anführungsstriche eingeschlossen sein müssen.

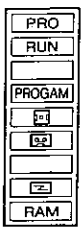
Die Zeichen, die den <Format-String> bilden, sind:

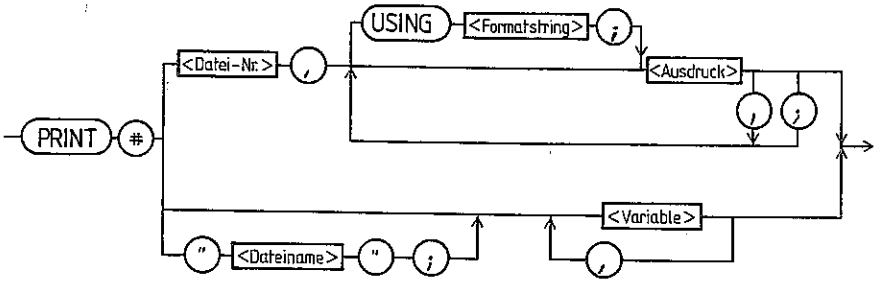
#	das Nummernzeichen,
&	das Kaufmannsund,
+	das Pluszeichen,
*	der Multiplikationsstern,
^	der Potenzoperator,
,	das Komma und
.	der Dezimalpunkt.

Das Nummernzeichen und das Kaufmannsund sind sogenannte Platzhalter. Für jedes im <Formatstring enthaltene # kann eine Ziffer des numerischen Wertes angezeigt werden, für jedes & dagegen ein Zeichen eines Strings. Alle weiteren Formatsymbole dienen der näheren Beschreibung numerischer Formate.

Bei den numerischen Formaten lassen sich sowohl positive als auch negative Werte darstellen. Das Vorzeichen wird jedoch (von einer Ausnahme abgesehen) nur bei den negativen Werten angezeigt.

Zusammen mit dem Stringformat lassen sich insgesamt acht grundlegende Formate unterscheiden, die in den Erklärungen zu USING näher erläutert sind.





Siehe auch : INPUT#, OPEN, PRINT USING

WIRKUNG : PRINT# und PRINT# USING werden benutzt, um Daten sequentiell in eine Datei zu schreiben.

HINWEISE : Disketten- oder RAM-Disk-Dateien

<Datei-Nr.> ist die Nummer, unter der die Datei mittels OPEN-Befehl geöffnet wurde. Der Versuch, etwas in eine ungeöffnete Datei zu schreiben bewirkt die Ausgabe eines ERROR-Codes.

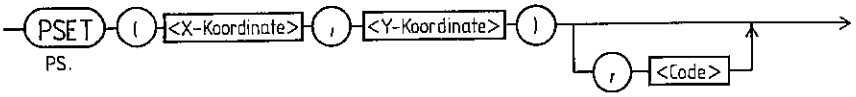
Sind die Daten durch Semikolons voneinander getrennt, so werden sie in der Datei ohne einen Zwischenraum aneinandergefügt. Wird ein Komma in der Auflistung der Daten verwendet, wird dagegen zwischen die Daten ein Leerzeichen geschrieben. Soll das Komma selbst mit in die Datei gelangen, ist es in Anführungsstriche zu setzen.

Die Benutzung von USING geschieht in gleicher Weise wie bei PRINT.

Cassetten-Dateien

Ist kein **<Dateiname>** spezifiziert, werden die Daten von der momentanen Bandposition an auf die Cassette aufgezeichnet. Als Seperator zwischen den einzelnen Daten kann bei Cassetten-Dateien lediglich das Komma verwendet werden.

Die Aufzählung der Daten kann sowohl numerische als auch String-Variablen enthalten. In beiden Fällen ist auf die Verwendung der richtigen Trennungszeichen zu achten. Ansonsten kann es Probleme beim Einlesen mittels INPUT# geben. Mit Array-Variablen können keine individuellen Elemente angesprochen werden. Es muß das gesamte Array in der Form A(*) angegeben sein.



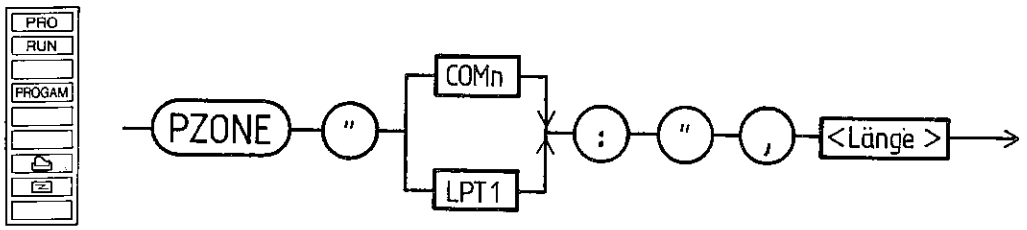
WIRKUNG : PSET setzt oder löscht einen Display-Punkt, der durch zwei Koordinaten bestimmt ist.

HINWEISE : Ein gesetzter Punkt erscheint dunkel, ein gelöschter dagegen hell.

Die Koordinaten X und Y können im Bereich von -32768 bis 32767 liegen, ohne daß ein ERROR-Code generiert wird. Es sollten jedoch die folgenden Bereiche beachtet werden, wenn ein tatsächlich existierender Display-Punkt anzusprechen ist:

<X-Koordinate> : 0...155
<Y-Koordinate> : 0....31

Benutzt man PSET ohne die Option <Code>, so wird der mit den Koordinaten bestimmte Display-Punkt gesetzt. Ist der <Code> vorhanden, invertiert PSET den gerade derzeitigen Zustand des Punktes. Ein heller Punkt wird dunkel, ein dunkler hell. Der <Code> muß aus dem Buchstaben X bestehen.



Siehe auch : LPRINT

WIRKUNG : PZONE bestimmt, innerhalb welcher Zonen die Ausgaben, die zum Drucker oder einem seriellen Port gehen, eingebettet werden.

HINWEISE : In BASIC werden die über LPRINT auszugebenden Daten in Zonen mit fester Zeichenanzahl ausgegedruckt bzw. ausgesendet. Diese in Zonen eingeteilte Ausgabe gilt dann, wenn die Daten in einer Liste durch Komma voneinander getrennt vorkommen. PZONE bestimmt die <Länge> der Zonen.

Diese zonenorientierte Ausgabe gilt auch dann, wenn die Daten über das Interface ausgesendet werden. Den rechtsbündig in der Zone liegenden Daten wird somit eine entsprechende Anzahl an Leerstellen (spaces) vorausgeschickt.

Folgende Schnittstellen sind ansprechbar:

COM1:	RS-232C-Interface
COM2:	SIO-Interface
COM:	das derzeit über SETDEV selektierte Interface

Folgende Werte sind für die <Länge> erlaubt:

Für COM1:	8...255	(Standard: 20)
Für COM2:	8....80	(Standard: 20)

BEISPIEL :

```
10:LET A=1,B=22,C=33
20:FOR I=1TO 30:LPRINT "-";:NEXT I
30:LPRINT
40:PZONE "LPT1:",10
50:LPRINT A,B,C
60:END
```

Siehe auch : DEGREE, GRAD

WIRKUNG : Versetzt den Computer in den Winkelmodus RADIAN.

HINWEISE : In diesem Modus werden alle Winkelangaben als im Bogenmaß gegeben angesehen und auch in diesem Maß ausgegeben.

Zur Kennzeichnung dieser Betriebsart wird in der Status-Zeile das Symbol RADIAN angezeigt.

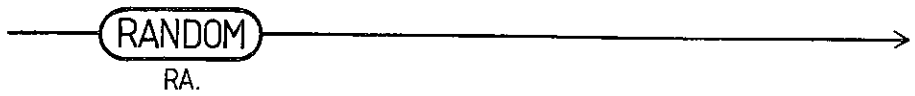
Die Argumente der Funktionen SIN, COS und TAN werden dann als im Bogenmaß vorliegend angesehen und die Werte der Funktionen ASN, ACS und ATN in diesem Maß geliefert.

BEISPIEL :

```
10:RADIAN
20:PAUSE "WINKEL IM BOGENMASS"
30:PRINT ASN(0.5),ASN(1)
40:PRINT ACS(0.5),ACS(1)
50:PRINT ATN(0.5),ATN(1)
60:END

>
RUN
WINKEL IM BOGENMASS
 5.235987E-01  1.570796327
  1.047197551             0
  0.463647609 7.853981E-01
>
```

Lassen Sie dieses Programm zum Vergleich auch in den beiden anderen Winkel-Modi (DEGREE und GRAD) laufen.



Siehe auch : RND

WIRKUNG : RANDOM bestimmt die Serie der Zufallszahlen, die über die Funktion RND geliefert werden.

HINWEISE : Benutzt man RANDOM zu Beginn eines Programmes, wird mit jedem Einschalten des Computer eine neue Reihe von Zufallszahlen erzeugt.

BEISPIEL :

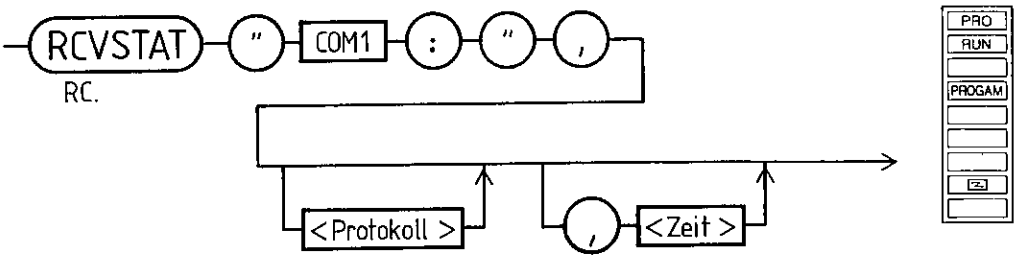
```
10:RANDOM
20:FOR I=1TO 5
30:PRINT I;" . ZUFALLSZAHL = ";
40:PAUSE RND(10)
50:NEXT I
60:END

>
RUN
1. ZUFALLSZAHL = 2
2. ZUFALLSZAHL = 1
3. ZUFALLSZAHL = 10
4. ZUFALLSZAHL = 8
5. ZUFALLSZAHL = 10
>

      OFF

      ON

>
RUN
1. ZUFALLSZAHL = 2
2. ZUFALLSZAHL = 1
3. ZUFALLSZAHL = 10
4. ZUFALLSZAHL = 8
5. ZUFALLSZAHL = 10
>
```



Siehe auch : RCVSTAT

ERKLÄRUNG : RCVSTAT setzt Steuersignale für das RS-232C-Interface und die Wartezeit (timeout) für beide seriellen Schnittstellen (SIO und RS-232C) für den Datenempfang.

HINWEISE : Die Auswahl des Interface erfolgt über den Parameter "COMn:". Für diesen gilt:

COM1:	RS-232C-Interface
COM2:	SIO-Interface
COM:	Zuletzt über SETDEV selektiertes Interface

Empfängt der PC-1600 von einem peripheren Gerät Daten über eines der seriellen Interfaces, so muß er, damit dieses einwandfrei funktioniert, stets vom Peripherigerät über spezielle Steuersignale darüber informiert werden, ob dieses senden möchte bzw. der PC-1600 dazu bereit ist, Daten zu empfangen. Diesen Austausch von Steuersignalen nennt man auch "Handshaking".

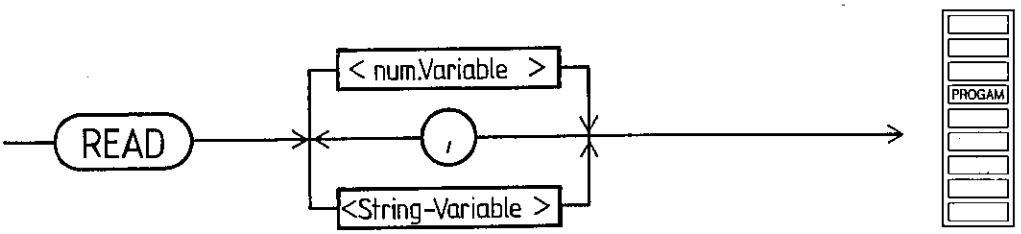
Die Zustände der Steuersignale, die für einen weiteren Datentransfer erfüllt sein müssen, werden durch den Parameter <Protokoll> bestimmt. Dieser Parameter ist eine Dezimalzahl im Bereich 0 ... 255 und wird intern in eine 8-Bit-Zahl (also ein Byte) umgewandelt. Jedes Bit bestimmt dabei den Zustand eines der speziellen Steuersignale. Sind alle Bedingungen erfüllt, kann der Datentransfer fortgesetzt werden. Die Bits 7 und 8 werden intern immer zu Null angenommen, da sie für das "Handshaking" nicht benötigt werden.

Die Zuordnung zwischen den Bits und den Signalen sowie die Bedeutung der Bitzustände lautet wie folgt:

Bit		Wert	Signal
(LSB)			
Niederwertigstes Bit	0	0	DTR high
		1	DTR low
	1	0	RTS high
		1	RTS low
	2	0	CTS high
		1	CTS low
	3	0	CD high
		1	CD low
	4	0	DSR high
		1	DSR low
	5	0	CI high
		1	CI low
	6	0	ungenutzt
Höchstwertigstes Bit (MSB)	7	0	ungenutzt

Fehlt der Parameter <Protokoll>, wird für ihn standardmäßig der Wert 63 (00111111) angenommen, was den Datenempfang ohne Steuersignale erlaubt.

Unter "timeout" versteht man die Zeitdauer, die der Computer längstens auf eine Antwort des Peripheriegerätes wartet, bis er den Datentransfer abbricht. Mit dem Parameter <Zeit> kann sie mit den Werten 1 bis 255 als ein entsprechend Vielfaches von 0.5 Sekunden eingestellt werden. Der Wert 0 setzt die Wartedauer auf unendlich, was auch der standardmäßigen Annahme entspricht, wenn dieser Parameter weggelassen wird.



Siehe auch : DATA, RESTORE

WIRKUNG : READ liest aus den im Programm befindlichen DATA-Zeilen die dort enthaltenen Werte ab und weist sie den gewünschten Variablen zu.

HINWEISE : Damit eine READ-Anweisung durchgeführt werden kann, muß innerhalb des Programmes mindestens eine DATA-Zeile vorhanden sein.

Jeder als Parameter aufgeführten Variablen wird dann der Reihe nach das nächste in einer DATA-Zeile auffindbare Element zugeordnet. Die in der Parameterauflistung enthaltenen Variablen müssen jeweils typenmäßig den in der DATA-Anweisung auftretenden Konstanten entsprechen.

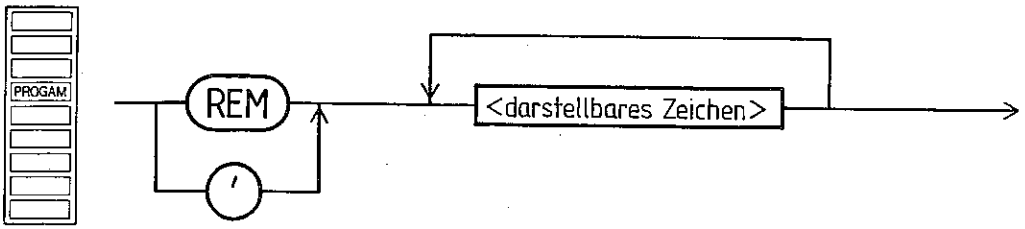
Die einzulesenden Werte brauchen nicht allesamt in einer gemeinsamen DATA-Zeile stehen, sondern können auf beliebig viele DATA-Zeilen verteilt sein. Es gilt dabei immer, daß mit der nächsten READ-Variablen der nächste in einer DATA-Zeile auffindbare Wert gelesen wird.

Sind die in den DATA-Zeilen stehenden Werte alle gelesen, führt die nächste READ-Anweisung zur Ausgabe eines ERROR-Codes.

Beachten Sie die Beschreibungen von MERGE, wenn Sie READ- und DATA-Anweisungen in Programmen benutzen, die mit MERGE verbunden werden sollen.

BEISPIEL :

```
10:DIM B(10)
20:FOR I=1 TO 10 : READ B(I)
30:PRINT B(I) : NEXT I
40:DATA 10,20,30,40,50
50:DATA 60,70,80,90,100
```



Siehe auch : LIST, LLIST

WIRKUNG : REM erlaubt die Einfügung von Kommentaren in den Programmtext.

HINWEISE : Diese Kommentare dienen zur Kennzeichnung von Programmteilen, so daß man ihre Funktionen noch zu einem späterem Zeitpunkt wieder erkennen und verstehen kann. Sie erscheinen ausschließlich im Programmlisting. Sie bleiben also bei der Programmausführung unsichtbar. Anstelle von REM kann auch der Apostroph ' verwendet werden.

Die so gekennzeichneten Programmzeilen zählen zu den nicht ausführbaren Anweisungen. Werden diese durch GOTO oder GOSUB angesprungen, erfolgt die Programmausführung mit der nächsten Zeile, die nicht als Kommentarzeile ausgewiesen ist.

Kommentare können an Anweisungen einer Zeile angefügt werden, wenn man vor dem Befehlswort REM den Doppelpunkt als Trennzeichen benutzt.

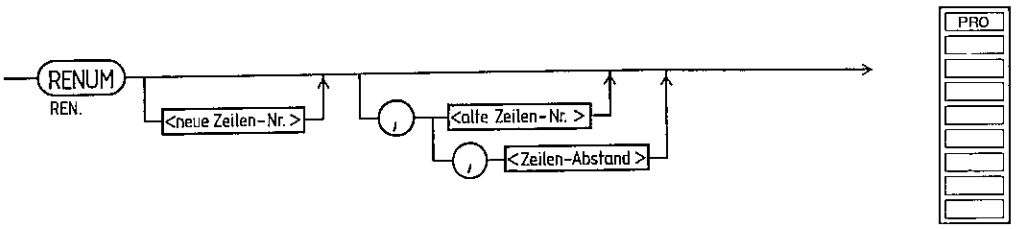
```
10:V=G*H/3 : REM VOLUMEN DER PYRAMIDE
```

Einem Apostroph darf kein Doppelpunkt vorausgehen, da dieser die Trennfunktion beinhaltet:

```
10:V=G*H/3 'VOLUMEN DER PYRAMIDE
```

Nach der Einleitung eines Kommentares gilt der gesamte Rest der Zeile als Kommentar. Hinter einem Kommentar stehende Anweisungen, die zur selben Zeile gehören, werden ignoriert:

```
10:REM VOLUMEN DER PYRAMIDE : V=G*H/3
20:'VOLUMEN DER PYRAMIDE : V=G*H/3
```



Siehe auch : DELETE, EDIT, LIST

WIRKUNG : RENUM nimmt eine Neunumerierung der Zeilen eines BASIC-Programmes vor.

HINWEISE : <alte Zeilen-Nr.> bestimmt, ab welcher Zeile mit der neuen Numerierung begonnen wird.

<neue Zeilen-Nr.> bestimmt, wie die erste neue Zeilennummer lautet.

<Zeilenabstand> bestimmt, in welchen Abstand die Nummern generiert werden sollen. Fehlt diese Angabe, erfolgt die Numerierung in Zehnerschritten.

Bei der Neunumerierung werden automatisch alle Zeilennummern in GOTO- und GOSUB-Anweisungen entsprechend korrigiert. Dies gilt auch für die Zeilennummern, die in anderen Verzweigungsanweisungen, wie z.B. IF..THEN..ELSE, enthalten sind. Taucht dabei aber eine nichtexistente Zeilennummer auf, wird die Meldung:

Undefined in <Zeilen-Nr.>

angezeigt und die Neunumerierung unterlassen. Die <Zeilen-Nr.> verweist dabei auf die Zeile, in der der Fehler entdeckt worden ist.

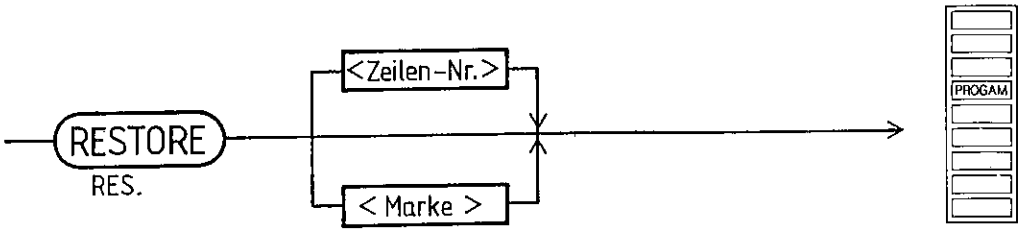
Die höchste generierbare Zeilennummer ist 65279.

BEISPIEL :

```
10:INPUT "ANTWORT (J/N) =";A$
20:IF A$="J" THEN 10
30:IF A$="N" THEN 60
40:PRINT "NUR MIT J ODER N ANTWORTEN !"
50:GOTO 10
60:END

>RENUM 100,10,5

100:INPUT "ANTWORT (J/N) =";A$
105:IF A$="J" THEN 100
110:IF A$="N" THEN 125
115:PRINT "NUR MIT J ODER N ANTWORTEN !"
120:GOTO 10
125:END
```



Siehe auch : READ, DATA

WIRKUNG : RESTORE setzt den internen DATA-Zeiger zurück oder auf den Anfang der gewünschten DATA-Zeile. Damit lassen sich die in den DATA-Zeilen bereitgestellten Werte erneut lesen.

HINWEISE : Wird RESTORE ohne Parameter verwendet, zeigt der interne Zeiger anschließend auf den ersten Wert der zuerst im Programm auffindbaren DATA-Zeile.

Bei der Angabe einer **<Zeilen-Nr.>** oder einer **<Marke>** wird der Zeiger auf das erste Element der in dieser Zeile vorkommenden DATA-Anweisung gesetzt.

Nach Setzung des Zeigers lassen sich die Werte der momentan vom Zeiger bestimmten DATA-Liste mit jeder folgenden READ-Anweisung der Reihe nach ablesen und den durch READ bestimmten Variablen zuweisen. Sind alle Werte einer DATA-Anweisung gelesen, sucht der Computer nach der nächsten im Programmspeicher befindlichen Zeile mit einer DATA-Anweisung und setzt den Zeiger auf deren Anfang und so fort.

Enthält die als Parameter angegebene Zeile keine DATA-Liste, wird der Zeiger auf den Anfang der nächsten auffindbaren DATA-Anweisung gesetzt.

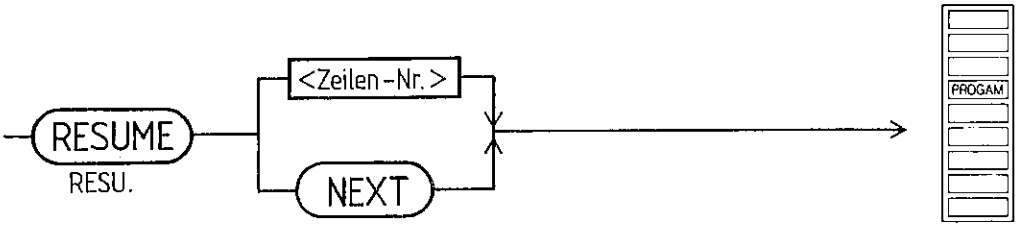
BEISPIEL :

```
100:DIM A$(3*10
110:GOSUB "OBST"
120:RESTORE
130:GOSUB "OBST"
140:RESTORE 310
150:GOSUB "OBST"
160:END
200:"OBST"
210:FOR N=1 TO 3
220:READ A$(I)
230:PAUSE A$
240:NEXT N
250:PAUSE
260:RETURN
300:DATA "PFLAUME","PFIRSICH","NEKTARINE"
310:DATA "APFEL","BIRNE","MANDARINE"


>
RUN
PFLAUME
PIRSICH
NEKTARINE

PFLAUME
PIRSICH
NEKTARINE

APFEL
BIRNE
MANDARINE
>
```



Siehe auch : ON ERROR GOTO, ERL, ERN

WIRKUNG : RESUME dient zur Rückkehr von einer Fehlerbehandlungs-Routine in das Hauptprogramm.

Das RESUME-Kommando wirkt ähnlich wie das RETURN eines normalen Unterprogrammes. Es darf aber nur in solchen Routinen verwendet werden, die der Auswertung und Behandlung von Fehlern dienen und von ON ERROR GOTO aufgerufen werden.

RESUME <Zeilen-Nr.>

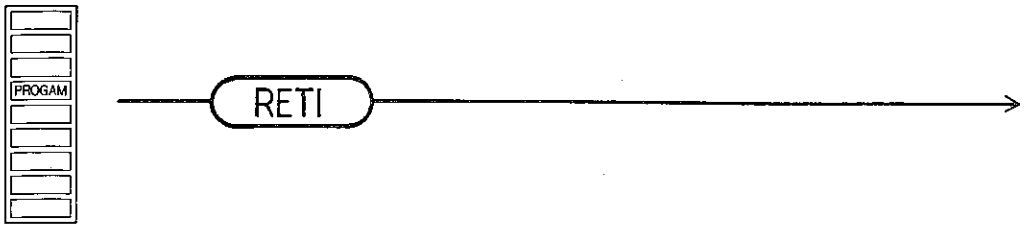
setzt das normale Programm nach Beendigung der Fehlerbehandlungsroutine mit der spezifizierten Zeile fort.

RESUME NEXT

setzt den normalen Programmablauf mit der Zeile fort, die unmittelbar auf diejenige folgt, die mit ihrem Fehler für die Aktivierung der Fehlerbehandlungsroutine sorgte.

RESUME

setzt den normalen Programmablauf mit der Zeile fort, die den Fehler verursachte.



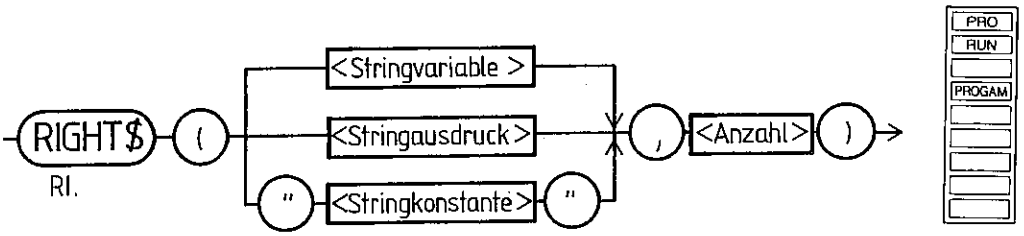
Siehe auch : ON ADIN GOSUB, ON COMn GOSUB, ON KEY GOSUB, ON PHONE GOSUB, ON TIME$ GOSUB

WIRKUNG : RETI dient zur Rückkehr von einer Interrupt-Routine in das Hauptprogramm.

HINWEISE : RETI wird in der gleichen Weise verwendet wie RETURN. Der Unterschied ist jedoch, daß RETI keine normalen Unterprogramme beendet, sondern solche, die der Behandlung eines Interrupts dienen. Solche Interrupt-Routinen werden durch die Anweisungen ON ADIN GOSUB, ON COMn GOSUB, ON KEYn GOSUB, OH PHONE GOSUB und ON TIME$ GOSUB aufgerufen. Sollte während der Bearbeitung einer Interrupt-Routine ein weiterer Interrupt angefordert werden, wird dieser mit Ausführung des RETI-Befehles angenommen und die entsprechende Interrupt-Routine bearbeitet.

BEISPIEL :

```
10:ON PHONE GOSUB
:
:
100:PRINT "DA IST EIN ANRUF FÜR DICH!"
110:RETI
```



Siehe auch : LEFT$, MID$

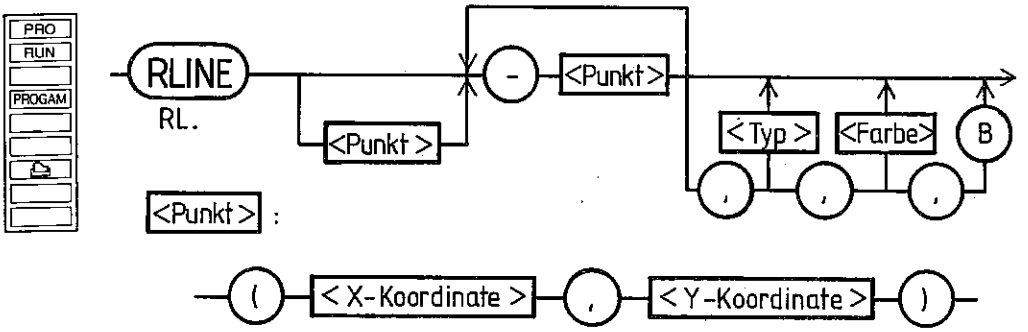
WIRKUNG : Die Funktion RIGHT$ liefert den rechtsbündigen Teil eines vorgegebenen Strings, wobei bestimmt werden kann, wieviele Zeichen von rechts aus dem String abgelesen werden.

HINWEISE : Die <Anzahl> der Zeichen des Teilstringes muß im Bereich von 0 bis 80 liegen. Gibt man die Anzahl als gebrochene Zahl an, wird sie zur nächsten ganzen Zahl hin gerundet. Ist die Anzahl größer als die Zeichenanzahl des vorgegebenen Strings, wird der gesamte vorgegebene String geliefert.

BEISPIEL :

```
5:WAIT 32
10:X$="SHARP PC-1600"
20:FOR N=1TO 14
30:S$=RIGHT$(X$,N)
40:PRINT S$
50:NEXT N
60:WAIT
70:END

>RUN
0
00
600
1600
-1600
C-1600
PC-1600
 PC-1600
P PC-1600
RP PC-1600
ARP PC-1600
HARP PC-1600
SHARP PC-1600
SHARP PC-1600
>
```



Siehe auch : COLOR, LLINE

WIRKUNG : RLINE druckt eine oder bis zu vier aufeinanderfolgende Linien zwischen den jeweils definierten relativen Koordinaten-Punkten.

HINWEISE : Im Grafik-Modus druckt RLINE eine Linie von der Koordinate (X1,Y1) oder der momentanen Position des Stiftes bis zum Punkt (X2,Y2). Die X- und Y-Koordinaten müssen im Bereich von -2048 bis 2047 liegen. Anders als bei LLINE beziehen sie sich hier auf die letzte Stift-Position und nicht auf den mit SORGN vereinbarten Koordinatenursprung.

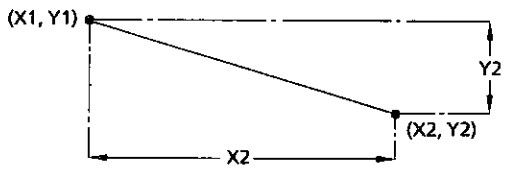
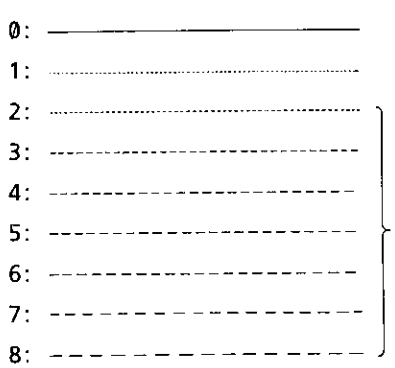


Abbildung 45

In einer RLINE-Anweisung können bis zu vier zu zeichnende Linien definiert werden.

Wird RLINE unmittelbar nach einem LPRINT-Befehl ausgeführt, muß der Parameter <Typ> unbedingt mit angegeben werden, da in diesem Falle für ihn kein Standardwert angenommen wird.

<Typ> bestimmt die aus 9 verschiedenen Mustern gewünschte Linienstruktur:



Der Typ 9 dient zum Test für die Stiftbewegung. Bei fehlender Angabe von <Typ> bleibt die letzte gewählte Einstellung gültig.

<Farbe> bestimmt die Wahl des Farbstiftes, von denen vier zur Auswahl stehen:

0	schwarz
1	blau
2	grün
3	rot

Bei fehlender Angabe der <Farbe> bleibt die zuletzt vorgenommene Einstellung erhalten.

Die Angabe der Option B sorgt für das Zeichnen eines Rechteckes, welches die Punkte (X1,Y1) und (X2,Y2) als Diagonale hat.

Die Anweisung RLINE (X1,Y1)-(X2,Y2),,,B ergibt also folgendes Bild:

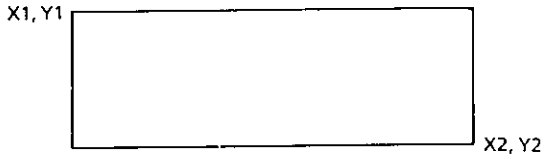
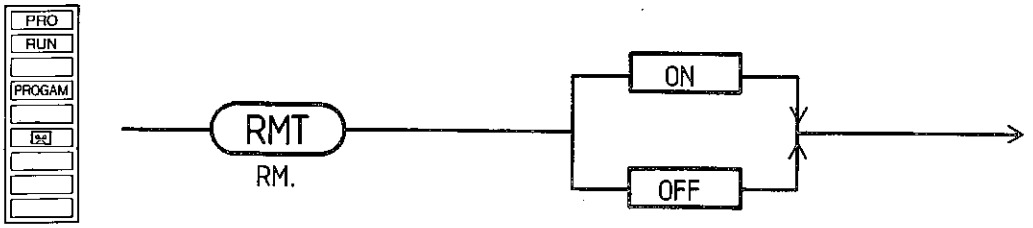


Abbildung 46

RMT ON/OFF



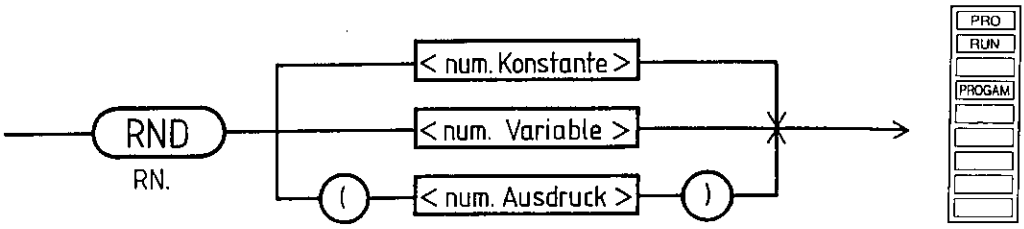
WIRKUNG : RMT ON/OFF gestattet oder verhindert die Fernbedienung des Cassetten-Recorder-Motors.

HINWEISE : RMT ON erlaubt die automatische Ein- bzw. Ausschaltung des Recorder-Motors bei Anwendung von BASIC-Befehlen, die sich auf Cassetten-Dateien beziehen.

RMT ON erlaubt die automatische Ein- bzw. Ausschaltung des Recorder-Motors bei der Anwendung von BASIC-Befehlen, die sich auf Cassetten-Dateien beziehen.

RMT OFF verbietet eine solche Fernbedienung des Recorder-Motors. Der Cassetten-Recorder läßt sich jedoch weiterhin manuell auf normale Weise bedienen.

Die automatische Steuerung des Motors kann nur dann erfolgen, wenn außer dem Befehl RMT ON auch der REMOTE-Schalter des Druckers aktiv ist, d.h. dieser in Stellung ON steht.



Siehe auch : RANDOM

WIRKUNG : Die Funktion RND liefert eine Zufallszahl, deren maximale Größe vom angegebenen Argument abhängt.

HINWEISE : Mit Hilfe der RND-Funktion lassen sich scheinbar willkürlich verteilte Zahlen erzeugen. Schaltet man jedoch den Computer an und startet das mit der RND-Funktion versehene Programm erneut, so stellt man fest, daß genau dieselben "Zufallszahlen" geliefert werden wie zuvor. Sollen die gelieferten Werte einem "echten Zufallsgesetz" folgen, muß vor Aufruf von RND-Funktionen im Programm das Befehlswort RANDOM aufgeführt sein.

Ist das Argument der RND-Funktion kleiner als Null liefert jeder erneute Funktionsaufruf immer wieder dieselbe Zufallszahl.

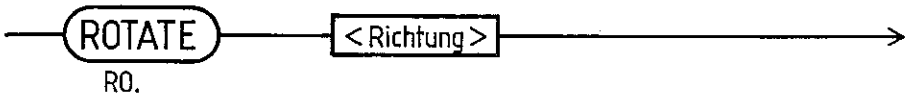
Ist das Argument größer als Null, aber kleiner als 1, liegen die gelieferten Funktionswerte im Bereich von 0 bis 1, ohne Einschluß der 1. Alle Werte werden hierbei mit bis zu 10 signifikanten Stellen geliefert.

Ist das Argument jedoch größer als 1, liegen die Funktionswerte im Bereich von 1 bis zum Argument selbst. Hierbei werden allerdings nur Integer-Werte ausgegeben.

BEISPIEL :

```
10:FOR I=1 TO 3
20:FOR  J=1 TO 10 : R=RND(9) : PRINT R; : NEXT J
30:PRINT : NEXT I
40:END

>RUN
6 4 2 5 6 8 2 7 6 8
5 5 7 7 1 2 6 5 3 6
3 1 5 7 3 4 5 7 4 2
```



WIRKUNG : ROTATE bestimmt die Orientierung der gedruckten Zeichen und die Druckrichtung.

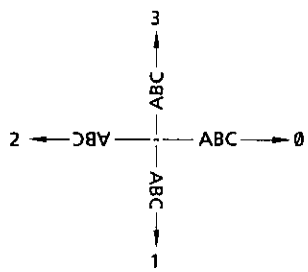
HINWEISE : Der Parameter **<Richtung>** bestimmt sowohl die Druckrichtung als auch die Orientierungslage der Zeichen. Er muß ganzzahlig sein und im Bereich von 0 bis 3 liegen. Diese Werte haben folgende Bedeutung:

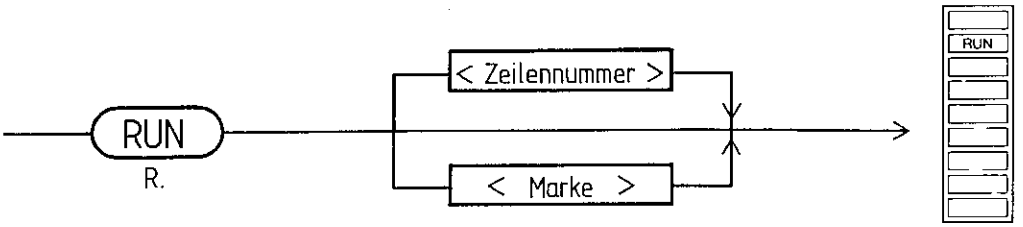
<Richtung>	Orientierung	Druckrichtung
0	normal lesbare Lage	nach rechts
1	auf rechter Seite liegend	nach unten
2	auf dem Kopf stehend	nach oben
3	auf linker Seite liegend	nach links

Entfällt der Parameter <Richtung>, bleiben alle bisherigen Einstellungen erhalten.

BEISPIEL :

```
5:GRAPH
10:GLCURSOR(400,0)
20:FOR D=0 TO 1
30:FOR P=0 TO 3
40:ROTATE P,D
50:LPRINT " ABC ";
60:NEXT P
70:NEXT D
80:END
```





Siehe auch : CONT, GOTO, LOAD, MERGE

WIRKUNG : RUN startet ein im Speicher befindliches BASIC-Programm.

HINWEISE : Ohne Parameterangabe beginnt die Ausführung des Programmes mit dessen erster Zeile, also der, der kleinsten vorkommenden Zeilennummer.

Mit Angabe der **<Zeilennummer>** oder einer **<Marke>** wird das Programm ab der spezifizierten Stelle getsartet.

In jedem Falle löscht RUN Variablen und setzt den internen Zeiger für die Auswahl der in den DATA-Zeilen bereitgestellten Daten auf die erste mögliche Position.

BEISPIELE :

```
>
RUN

>
RUN 100

>
RUN "F"

>
RUN "MARKE"
```

WIRKUNG : RXD$ liefert die Daten, sofern vorhanden, die über das momentan selektierte Interface an den Computer abgegeben werden.

HINWEISE : Das derzeit am Interface empfangene Daten-Byte wird von RXD$ als String geliefert.

Ist momentan kein Zeichen am selektierten Port (Interface) vorrätig, so erhält man über RXD$ zwei Leerzeichen (jeweils Code &20H).

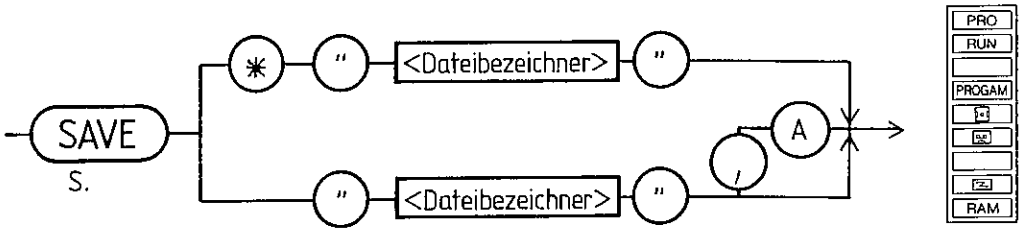
Liegt ein Übertragungsfehler am Interface vor, liefert RXD$ die Code-Folge &3F,&20,&20, also ein Fragezeichen mit zwei nachfolgenden Spaces.

BEISPIEL :

```
10:FOR I=1 TO 4
20:A$=A$+"*"+RXD$
30:NEXT I
40:PRINT A$
50:END
```

Lassen Sie dieses Programm laufen, ohne daß an eine der seriellen Schnittstellen etwas angeschlossen ist. Somit können über diese folglich auch keine Daten ankommen. RXD$ muß daher bei jedem Aufruf zwei Leerstellen liefern. Damit ist das Ergebnis des Programmablaufes wie folgt:

```
>RUN
*  *  *  *
>
```



Siehe auch : LOAD, MERGE

WIRKUNG : SAVE sichert eine Datei auf einer Diskette oder RAM-Disk oder sendet diese über ein serielles Interface aus.

HINWEISE : SAVE sichert die Datei unter dem angegebenen Namen auf dem spezifizierten Medium oder sendet es über das gewählte Interface aus.

SAVE* kann nur die mit dem Befehl REM oder einem Apostroph gekennzeichneten Kommentarzeilen eines BASIC-Programmes sichern. Ansonsten verhält sich dieser Befehl wie SAVE. Auf diese Weise lassen sich Text-Dateien abspeichern, die später mit LOAD* wieder geladen werden können.

Gibt man die Option A als Parameter an, erfolgt die Dateisicherung im ASCII-Format. Fehlt diese Angabe, wird das kompaktere Binärformat gewählt.

Existiert bereits eine Datei unter gleichem Namen, so wird diese mit der neuen Datei überschrieben, sofern sie nicht durch die P-Option des SET-Kommandos geschützt worden ist.

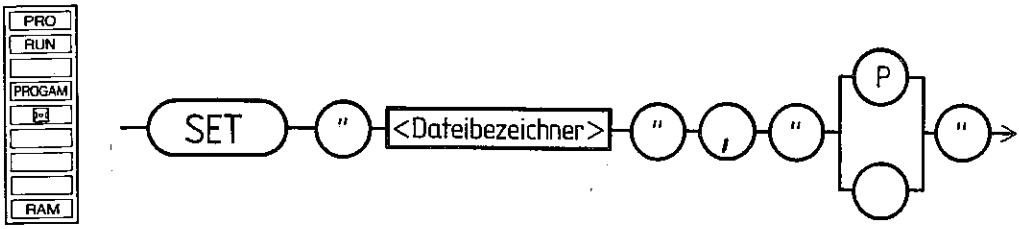
Bei RAM-Modulen ist ein Überschreiben dann nicht möglich, wenn sich der Schreibschutzschalter in Stellung ON befindet.

Läßt man in der Dateibezeichnung die Extension weg, wird sie automatisch zu .BAS angenommen.

BEISPIELE :

```
>
SAVE "X:CHESS"

>
SAVE* "S2:INFO"
```



WIRKUNG : SET setzt oder entfernt den Schreibschutz von Disketten- oder RAM-Disk-Dateien.

HINWEISE : Das SET-Kommando dient dem Schutz von Dateien gegen unerlaubten Zugriff bzw. unbeabsichtigter Löschung.

Der Parameter **<Dateibezeichner>** spezifiziert das Speichermedium, den Dateinamen und die Extension der zu schützenden Datei.

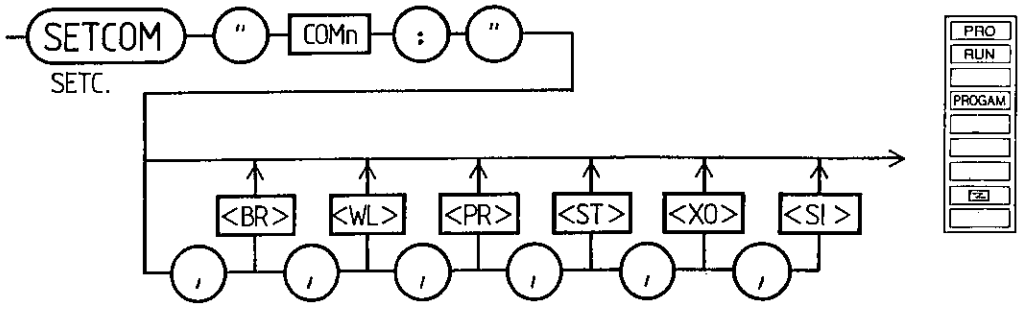
"P" setzt den Schreibschutz.

Die betreffende Datei kann danach weder:

- mit dem NAME-Kommando umbenannt werden, noch
- mit dem KILL-Kommando gelöscht werden, noch
- im APPEND- oder OUTPUT-Modus geöffnet werden.

Es sind keine Daten in eine mit "P" geschützte Datei schreibbar.

" " beseitigt den Schreibschutz wieder.



Siehe auch : COM$, SETDEV

WIRKUNG : SETCOM definiert das Kommunikationsprotokoll der seriellen Schnittstellen.

HINWEISE : Folgende Interfaces können gewählt werden:

```
COM1:    RS-232C-Interface
COM2:    SIO-Interface
COM:     mit SETDEV selektiertes Interface
```

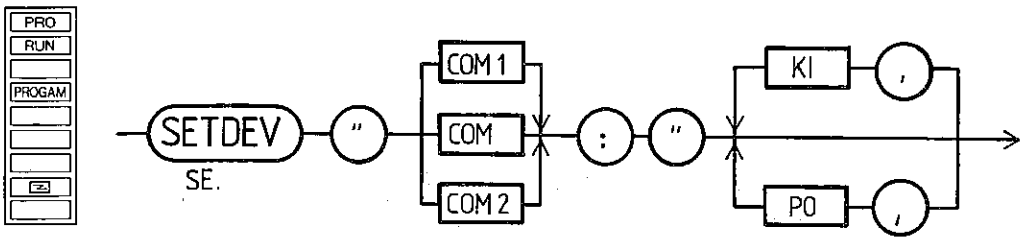
Das Kommunikationsprotokoll wird durch sechs optionale Parameter, die durch Kommas voneinander getrennt sein müssen, bestimmt. Dabei gilt:

```
<BR>  Übertragungsrate    :  50 bis 38400 baud
<WL>  Wortlänge           :   5 bis 8 Bits
<PR>  Parität             :   E (even) = gerade
                              O (odd)  = ungerade
                              N (none) = keine
<ST>  Stopbit-Anzahl      :   1 oder 2
<XO>  XON/XOFF            :   X = ja; N = nein
<SI>  SHIFT IN/SHIFT OUT  :   S = ja; N = nein
      (Nur für 7-Bit-Übertragung)
```

Die standardmäßigen Einstellungen lauten:

```
RS-232C :  1200,8,N,1,X,S
SIO     : 38400,7,E,2,X,S
```

Werden die Befehle SAVE, LOAD, BSAVE oder BLOAD benutzt, um binäre Dateien über einen seriellen Port zu übertragen, muß die Wortlänge auf 8 Bit und der Parameter <SI>=N gesetzt werden. Bei der Sicherung von ASCII-Dateien mittels SAVE oder dem Laden solcher Dateien über LOAD gelten keine Einschränkungen der Kommunikationsparameter. Dieses gilt auch für die Verwendung der Befehle PRINT# und INPUT#.



WIRKUNG : SETDEV bestimmt, welches serielle Interface für die Ein- oder Ausgabe von Daten bei Anwendung bestimmter BASIC-Befehle zu verwenden ist.

HINWEISE : Das Kommando SETDEV öffnet damit im Grunde ein Interface für den Datentransfer. Die Selektion übernehmen folgende Parameter:

COM1: öffnet das RS-232C-Interface
COM2: öffnet das SIO-Interface
COM: beläßt das zuletzt verwendete Interface

Mit zwei zusätzlichen Parametern, die optional sind und in beliebiger Reihenfolge angegeben werden können, lassen sich die Ein- und Ausgabe von Daten wie folgt leiten:

PO Leitet die über LPRINT, LLIST und LFILES gelieferten Daten an das spezifizierte Interface.

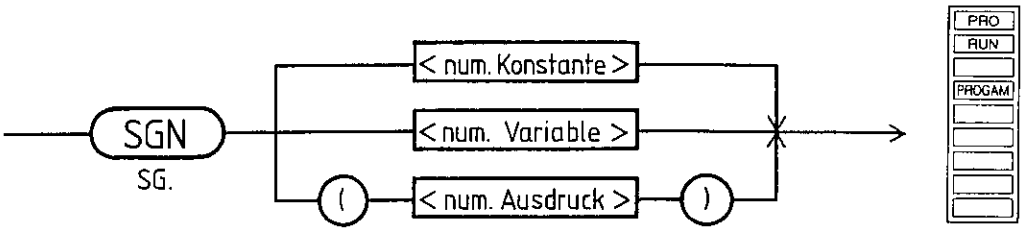
KI Bestimmt, daß der INPUT-Befehl die Daten vom spezifizierten Interface holt.

Wird der Befehl SETDEV ohne jegliche Angabe von Parametern verwendet, werden die standardmäßigen Einstellungen wieder hergestellt. Alle Datenausgaben werden damit zum Drucker geführt und die Eingaben von der Tastatur erwartet.

BEISPIELE :

```
>
SETDEV "COM1:",PO

>
SETDEV
```



WIRKUNG : Die Funktion SGN(X) liefert das Vorzeichen des Argumentes X.

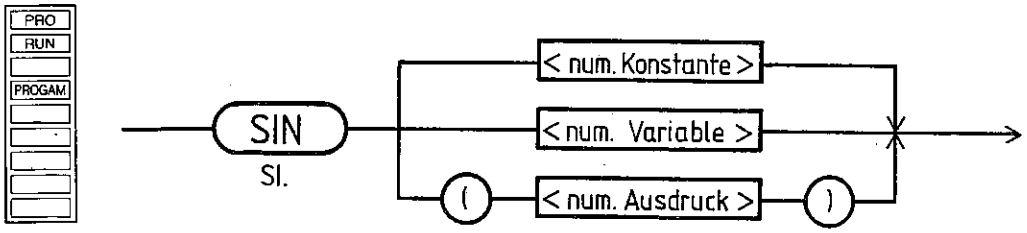
HINWEISE : X kann dabei jeder beliebige numerische Ausdruck sein. Die dazu gelieferten Funktionswerte lauten -1,0 oder 1 und ergeben sich wie folgt:

Argument X	Funktionswert SGN(X)
X > 0	1
X = 0	0
X < 0	-1

BEISPIEL :

```
5:WAIT 100
10:FOR N=-3 TO 3
20:PRINT N,SGN(N)
30:NEXT N
40:END

>
RUN
              -3            -1
              -2            -1
              -1            -1
               0             0
               1             1
               2             1
               3             1
>
```



Siehe auch : ASN, COS, TAN

WIRKUNG : Diese Funktion liefert den Sinus-Wert eines Winkelargumentes.

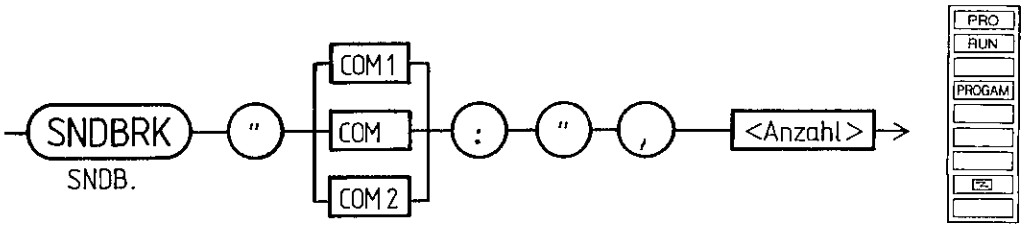
HINWEISE : Der angegebene Winkel kann in Altgrad, Bogenmaß oder Neugrad vorliegen. Damit der richtige Wert der Sinusfunktion geliefert werden kann, muß der Computer in den richtigen Winkelmodus geschaltet sein. Hierbei gilt:

Winkelmaß	Winkelmodus
Altgrad	DEGREE
Bogenmaß	RADIAN
Neugrad	GRAD

BEISPIEL :

```
10:DEGREE
20:G$=CHR$ (&F8)
30:PRINT "sin(30";G$;") = ";SIN(30)
40:PRINT "sin(45";G$;") = ";SIN(45)
50:END

>
RUN
sin(30°) = 0.5
sin(45°) = 7.071067812E-01
>
```



WIRKUNG : SNDBRK sendet eine vereinbarte Anzahl von Unterbrechungscodes (break codes) über das gewählte Interface aus, um dem sendenden Peripheriegerät mitzuteilen, daß momentan keine Daten angenommen werden können.

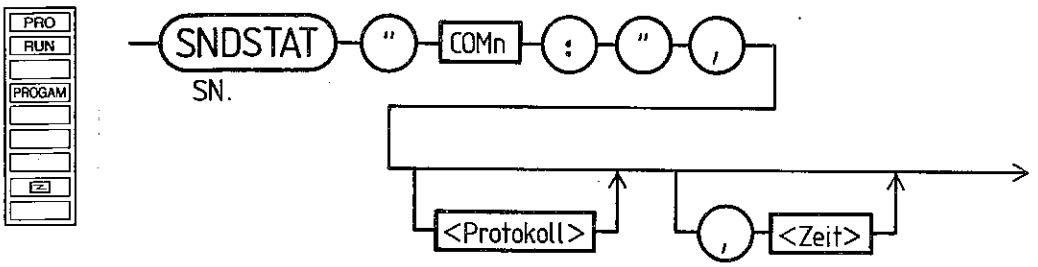
HINWEISE : Das Interface wird wie folgt ausgewählt:

COM1:	RS-232C-Interfac
COM2:	SIO-Interface
COM:	über SETDEV selektiertes Interface

Der Parameter **<Anzahl>** bestimmt, wie oft der Unterbrechungs-Code hintereinander ausgesendet werden soll. Der betreffende Wert kann zwischen 1 und 255 liegen.

BEISPIEL :
```
>
SNDBRK "COM1:",20
```

SNDSTAT



Siehe auch : RCVSTAT

WIRKUNG : SNDSTAT bestimmt das "Handshake"-Protokoll des RS-232C-Interface und die Wartezeit (timeout) beider serieller Schnittstellen.

HINWEISE : Die Auswahl der Schnittstelle erfolgt über den Parameter "COMn:", für den gilt:

COM1:	RS-232C-Interface
COM2:	SIO-Interface
COM:	über SETDEV selektiertes Interface

Sendet der PC-1600 an ein peripheres Gerät Daten über eine der seriellen Schnittstellen aus, so muß er stets vom Peripherigerät über spezielle Steuersignale darüber informiert werden, ob er weiter senden darf oder nicht. Diese Prozedur nennt man "Handshaking".

Die Steuersignal-Zustände, die zur Fortsetzung des Datentransfers erfüllt sein müssen, werden mit dem Parameter **<Protokoll>** festgelegt, der durch eine Dezimalzahl im Bereich von 0 bis 255 gegeben sein kann. Dieser wird intern durch eine 8 Bit umfassende binäre Zahl dargestellt. Jedes Bit bestimmt dabei den geforderten Zustand eines speziellen Steuersignales. Sind alle Bedingungen erfüllt, kann der Datentransfer weitergeführt werden. Die Bits 7 und 8 sind für den Handshake nicht maßgebend und werden daher stets zu Null angenommen.

Die Zuordnung zwischen den Bits und den Signalen sowie die Bedeutung der Bitzustände lautet:

Bit		Wert	Signal
(LSB)			
Niederwertigstes Bit	0	0	DTR high
		1	DTR low
	1	0	RTS high
		1	RTS low
	2	0	CTS high
		1	CTS low
	3	0	CD high
		1	CD low
	4	0	DSR high
		1	DSR low
	5	0	CI high
		1	CI low
	6	0	ungenutzt
Höchstwertigstes Bit (MSB)	7	0	ungenutzt

Fehlt der Parameter <Protokoll>, wird für diesen standardmäßig der Wert 59 (00111011) angenommen. In diesem Falle ist das Signal CTS also high.

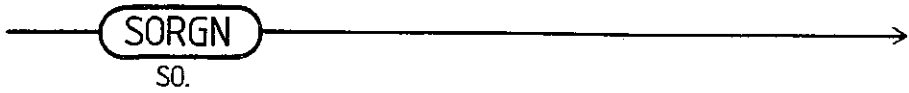
Unter "timeout" versteht man die Zeitdauer, die der Computer längstens auf eine vom Peripheriegerät stammende Antwort wartet, bevor er den Datentransfer abbricht. Mit dem Parameter <Zeit> kann diese als ein Vielfaches von 0.5 Sekunden eingestellt werden, wobei die Parameterwerte 1 bis 255 zulässig sind.

Der Wert 0 setzt die Wartedauer auf unendlich, was der standardmäßigen Einstellung entspricht, wenn dieser Parameter weggelassen wird.

BEISPIEL :

```
>
SNDSTAT "COM1:",45
```

SORGN



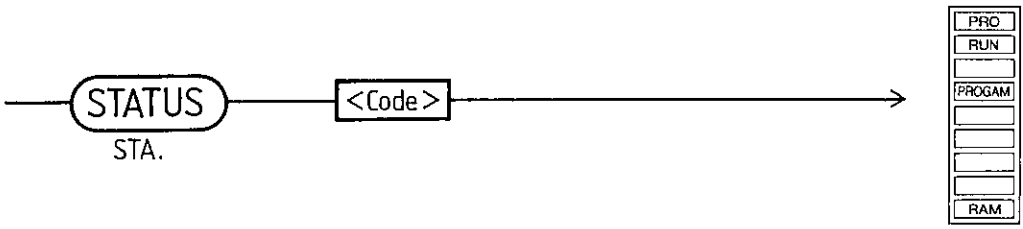
Siehe auch : LLINE, GLCURSOR

WIRKUNG : SORGN bestimmt die momentane Stiftposition des Druckers als künftigen Koordinatenursprung.

HINWEISE : Diese Festlegung des Koordinatenpunktes (0,0) an die derzeitige Stiftposition gilt nur für den Grafik-Modus und bezieht sich damit nur auf alle folgenden grafischen Befehle.

Mit SORGN kann der standardmäßige Koordinatenursprung nach den individuellen Bedürfnissen verlagert werden.

Die standardmäßige X-Koordinate des Koordinatenursprungs wird durch die am linken Papierrand liegende Schreibposition bestimmt, die Y-Koordinate durch die mit den Tasten [↑] und [↓] setzbare Papierposition.



Siehe auch : MEM, NEW

WIRKUNG : STATUS ermittelt die Anzahl der freien Speicherplätze an und liefert weitere Informationen über andere Speicherbereiche.

HINWEISE : Der Parameter **<Code>** bestimmt denjenigen Bereich des Speichers, über den nähere Informationen gewünscht sind und welcher Art diese sein sollen. Hierbei gilt folgender Zusammenhang:

<Code>	Information
0	Anzahl der freien Speicherplätze des Benutzerbereiches gemessen in Bytes. (Freier Bereich + Bereich der Variablen)
1	Größe des momentan geladenen Programmes in Bytes.
2	Niederwertigste Speicher-Adresse, ab der der freie Benutzerbereich beginnt.
3	Höchstwertigste Speicher-Adresse, mit der der freie Benutzerbereich endet.
4 : : : 255	Letzte Zeilennummer des auszuführenden BASIC-Programmes.

256 Nummer der Speicherbank, in der die niederwertigste Adresse des freien Benutzerbereiches existiert.

257 Nummer der Speicherbank, in der die höchstwertigste Adresse des freien Benutzerbereiches existiert.

258 Betrag (in Bytes) des nichtzugewiesenen Speicherplatzes innerhalb des freien Benutzerbereiches.

259 Anzahl der freien Bytes des im Modulfach S1 steckenden Programm-Modules, die von dessen Programm nicht belegt sind.

260 Anzahl der freien Bytes des im Modulfach S2 steckenden Programm-Modules, die von dessen Programm nicht belegt sind.

Speicheraufteilung des internen RAMs:

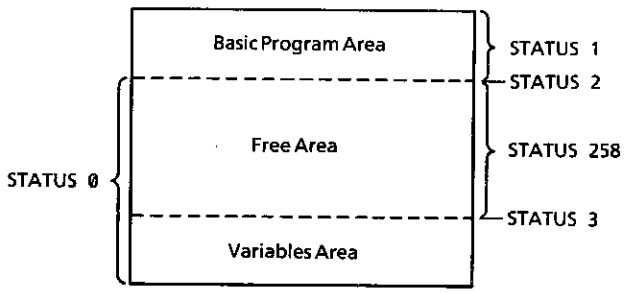


Abbildung 49

Speicheraufteilung des internen RAMs und der Erweiterungs-Module:

Das folgende Bild zeigt die Speicheraufteilung bei Verwendung von:

a) einem CE-1600M-Modul im Fach S1, wobei es für die Aufnahme von Programmen vorgesehen ist und der davon nicht berührte Speicherplatz zur Erweiterung des Arbeitsspeichers genutzt wird,

b) einem CE-1600M-Modul in Fach S2, welches ausschließlich zur Programmspeicherung initialisiert worden ist.

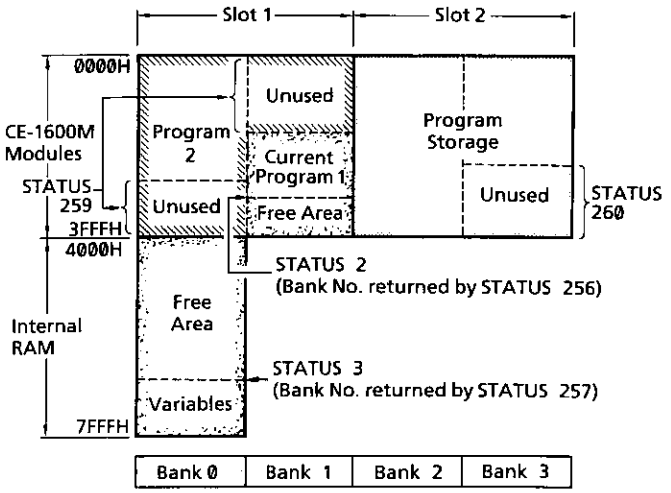
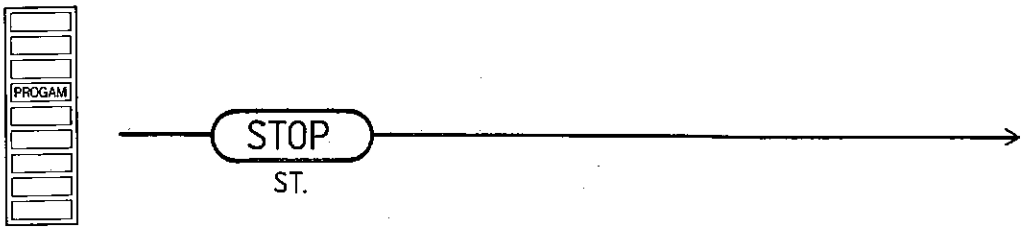


Abbildung 50 : Speicheraufteilung



Siehe auch : CONT, END

WIRKUNG : STOP dient der Unterbrechung eines Programmes, um so während der Testphase bei fehlerhaften Programmteilen anhalten und diese untersuchen zu können.

HINWEISE : Kommt ein STOP-Befehl zur Ausführung, wird das laufende Programm abgebrochen und eine Meldung angezeigt, die angibt, bei welcher Zeile dieser Abbruch erfolgte:

BREAK IN <Zeilennummer>

Nun kann durch Abfrage der Variablen und anderer Untersuchungen auf die Ursache der ermittelten Programmfehler geschlossen werden. Ebenso lassen sich nun den falsch belegten Variablen über die Tastatur die erwarteten Werte zuweisen und durch Start des Programmes mit der Anweisung CONT das weitere Verhalten getestet werden. Diese Fortsetzung ist aber nur möglich, wenn inzwischen keine Änderungen der Programmzeilen vorgenommen worden sind.

Im Gegensatz zum Befehl END werden bei STOP keine geöffneten Dateien geschlossen.

BEISPIEL :

```
10:FOR N=1 TO 10
20:LET S=N*5
30:STOP
40:GRAPH
50:LINE (0,0)-(N,S)
50:NEXT N
```

Da der Computer nur blockweise kopieren kann, mag es bei sehr umfangreichen Dateien vorkommen, daß Sie zu einem mehrfachen Wechsel zwischen Quell- und Zieldiskette aufgefordert werden. Ist der Sicherungsvorgang abgeschlossen, stoppt das Laufwerk und die grüne Lampe erlischt. Ebenso erscheint dann auch das Bereitschaftszeichen > wieder auf dem Display. Sie können nun die Sicherungsdiskette entnehmen, aufbewahren und den PC-1600 für andere Dinge nutzen.

Weitergehende Informationen erhalten Sie aus den Erklärungen der Befehle COPY und FILES, die im Kapitel 14 gegeben sind.

7) Referenz-Hinweise

- o Die Bedienungsanleitungen des Druckers CE-1600P und des Disketten-Laufwerks CE-1600F liefern Ihnen Informationen über:

  - Anschluß des Laufwerkes,
  - Einlegen und Entnahme von Disketten,
  - Schutz der Disketten vor versehentlichem Beschreiben,
  - Spezifikationen und
  - Grundsätzliche Vorsichtsmaßnahmen und Bedienungshinweise

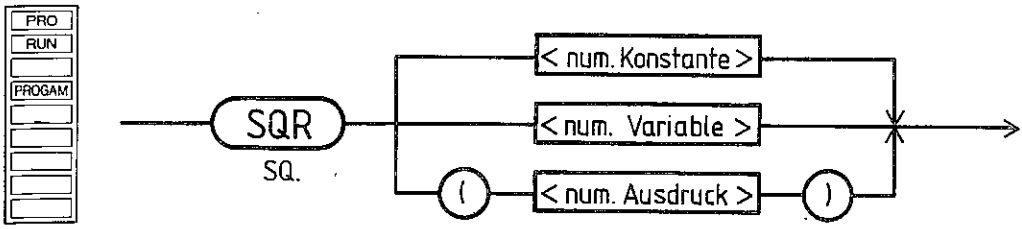
- o Dieses Handbuch gibt Auskunft über:

  - die BASIC-Befehle zum Schreiben und Lesen von Daten,
  - die dabei möglicherweise auftretenden ERROR-Codes und
  - die Erzeugung und Änderung von Dateien in BASIC.

b) Disketten-Befehle

Die im folgenden aufgelisteten BASIC-Befehle erlauben Ihnen das Lesen von Disketten-Dateien oder das Schreiben derselben. Genaue Erklärungen hierzu finden Sie in Kapitel 14.

BLOAD	Lädt ein Maschinensprache-Programm von der Diskette in den Arbeitsspeicher des Computers.
BSAVE	Schreibt ein Maschinensprache-Programm auf Diskette.
CLOSE	Schließt eine Datei und beendet so die Möglichkeit, auf diese zugreifen zu können.
COPY	Kopiert eine Datei unter anderem Namen auf demselben Speichermedium oder unter beliebigem Namen von einem Medium auf ein anderes.



WIRKUNG : Die Funktion SQR(X) liefert den positiven Wert der Quadratwurzel von X.

HINWEISE : Das Argument X darf ein beliebiger numerischer Ausdruck sein, dessen Wert jedoch im positiven Zahlen-Bereich liegen oder Null lauten muß. Negative Werte sind nicht erlaubt und führen daher zur Ausgabe eines ERROR-Codes.

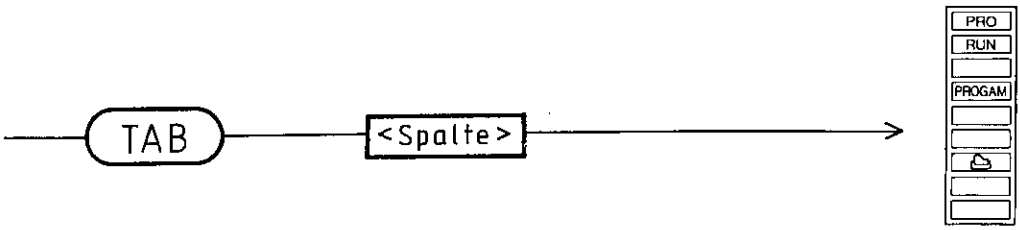
Anstelle des funktionalen Operators SQR ist auch die Verwendung des Wurzelzeichens √ möglich. Dieses Zeichen ist in MODE 1 durch die offene eckige Klammer [ersetzt.

BEISPIELE :

```
>
SQR(5)
                    2.236067977
>


>
MODE 1
√2
                    1.414213562
>


>
MODE 0
[3
                    1.732050808
>
```



Siehe auch : LCURSOR, LPRINT

WIRKUNG : TAB bewegt im Text-Modus den Druckstift an die angegebene Druckspalte.

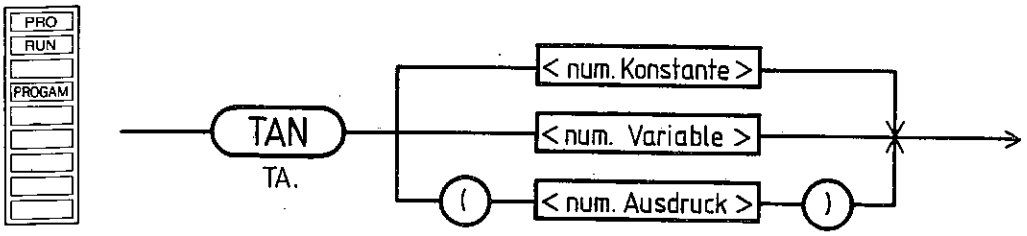
HINWEISE : Das TAB-Kommando verhält sich ähnlich wie das Kommando LCURSOR. Es bewegt den Druckstift an die angegebene Druckspalte, die im Bereich von 0 bis zum mit PCONSOLE festgelegten Maximalwert liegen kann.

Die Spaltenposition ist abhängig von der mit CSIZE eingestellten Zeichengröße.

TAB kann auch innerhalb einer LPRINT-Anweisung stehen. Im Gegensatz zu anderen BASIC-Versionen ist TAB aber nicht mit dem normalen PRINT-Befehl verwendbar.

BEISPIEL :

```
10:TEXT
20:LPRINT "LINKS";
30:TAB (38)
40:LPRINT "MITTE";
50:LPRINT TAB (70);"RECHTS"
60END
```



Siehe auch : ATN, COS, SIN

WIRKUNG : Die Funktion TAN liefert den Tangens eines als Winkel aufzufassenden numerischen Wertes.

HINWEISE : Der angegebene Winkel kann in Altgrad, Bogenmaß oder Neugrad vorliegen. Damit der richtige Wert der Tangensfunktion geliefert wird, muß sich der Computer im entsprechenden Winkelmodus befinden:

Winkelmaß	Winkelmodus
Altgrad	DEGREE
Bogenmaß	RADIAN
Neugrad	GRAD

Da die Tangensfunktion sogenannte Polstellen aufweist, an denen der Wert ins Unendliche geht, wird bei diesen Winkelargumenten (z.B. 90°) ein ERROR-Code ausgegeben.

BEISPIEL :

```
10:DEGREE
20:PAUSE "WINKEL SIND IN ALTRGRAD !"
30:PAUSE "WINKEL:  0,  TANGENS:";TAN(0)
40:PAUSE "WINKEL: 45,  TANGENS:";TAN(45)
50:PAUSE "WINKEL: 90,  TANGENS:";TAN(90)
```

```
>RUN
WINKEL SIND IN ALTGRAD !
WINKEL:  0,  TANGENS:  0
WINKEL: 45,  TANGENS:  1
WINKEL: 90,  TANGENS:
ERROR 37 IN 70              (Taste [CL] drücken !)
```

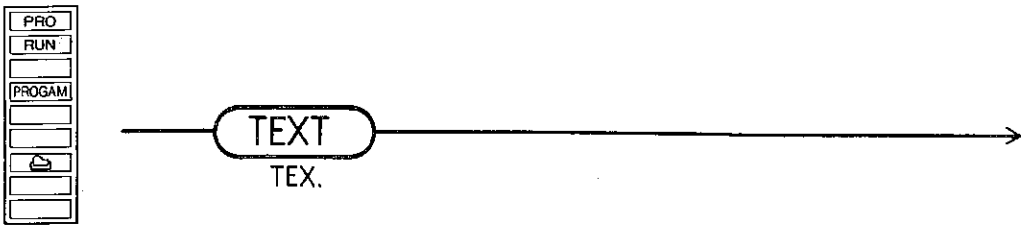
WIRKUNG : TEST prüft, ob alle vier Druckstifte funktions-funktionsfähig sind, indem mit jeder Farbe vier kleine Quadrate beschrieben werden.

HINWEISE : Die Quadrate werden nacheinander mit folgenden Farben gezeichnet:

schwarz, blau, rot und grün.

Anschließend wird der Drucker automatisch in den TEXT-Modus versetzt.

Um den Test zu unterbrechen, ist die BREAK-Taste zu betätigen.

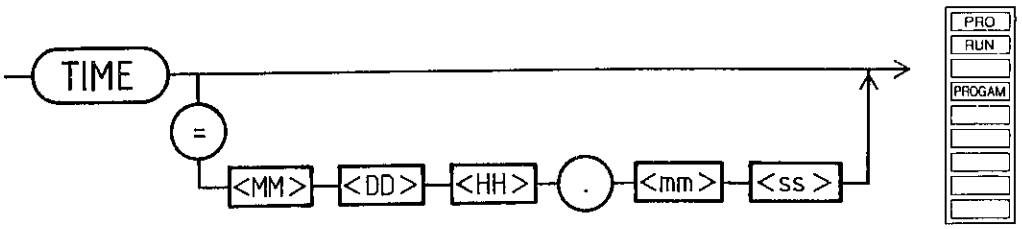


Siehe auch : GRAPH

WIRKUNG : TEXT versetzt den Drucker in den Text-Modus.

HINWEISE : Bei Aktivierung des Text-Modus wird die Größe der Zeichen auf den Wert 2 von CSIZE eingestellt und der Koordinatenursprung auf die dem linken Rand der momentanen Zeile am nächsten liegenden Position festgelegt.

TEXT setzt die mit den Parametern <Limit 1> und <Limit 2> gemachten Einstellungen des Befehles PAPER auf die Standardwerte zurück.



Siehe auch : TIME$

WIRKUNG : TIME$ setzt oder liefert Datum und Uhrzeit der im Computer eingebauten Echtzeit-Uhr.

HINWEISE : TIME ohne Parameterangabe liefert das momentane Datum und die momentane Uhrzeit im nachstehenden Format:

MMDDHH.mmss

Dabei bedeuten die hier gezeigten Platzhalter folgendes:

MM	Monat	01 bis 12
DD	Tag	01 bis 31
HH	Stunde	00 bis 23
mm	Minute	00 bis 59
ss	Sekunde	00 bis 59

Um Datum und Uhrzeit zu setzen, müssen diese im eben gezeigten Format der numerischen Variable TIME zugewiesen werden:

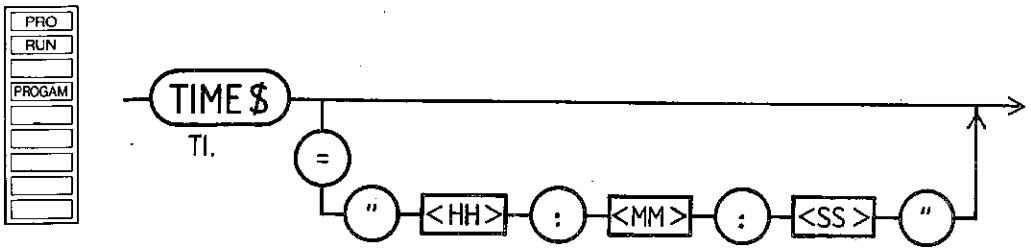
```
TIME = MMDDHH.mmss
```

Die Echtzeit-Uhr läuft auch dann weiter, wenn der Computer ausgeschaltet ist. Da das Datum ohne die Jahresangabe erfolgt, kann somit bei den Monatstagen der Einfluß eines Schaltjahres nicht berücksichtigt werden.

BEISPIEL :

```
>
TIME=031220.3215
```

Diese Anweisung stellt die Echtzeit-Uhr auf 20 Uhr, 32 Minuten und 15 Sekunden des 12. März.



Siehe auch : DATES$, TIME$ ON/OFF/STOP, ON TIME$ GOSUB

WIRKUNG : TIME$ ist eine Systemvariable, die die momentane Uhrzeit der im Computer eingebauten Echtzeit-Uhr enthält.

HINWEISE : TIME$ ohne Parameterangabe liefert einen String, der die Uhrzeit im folgenden Format angibt:

HH:MM:SS

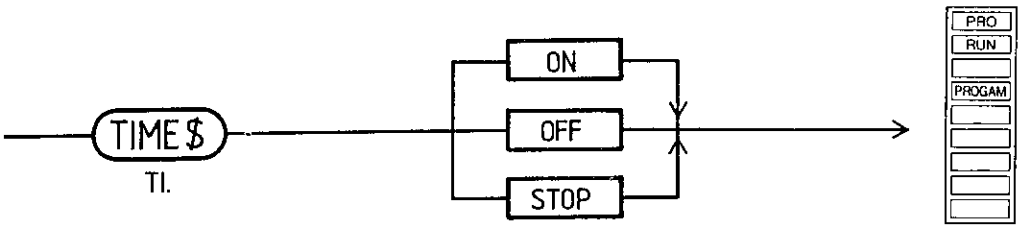
Dabei bedeuten: HH die Stunden von 00 bis 23
MM die Minuten von 00 bis 59
SS die Sekunden von 00 bis 59

Die Uhrzeit kann gestellt bzw. gesetzt werden, indem man der Systemvariablen TIME$ einen String im eben beschriebenen Format zuweist:

TIME$="HH:MM:SS"

BEISPIEL :

```
>
TIME$="09:15:00"          Uhrzeit setzen
09:15:00
>
:
:                         Geraume Zeit warten
:
TIME$                     Zeit abfragen
09:15:06                  Aktuelle Zeitangabe
>
```



Siehe auch : TIME$, ON TIME$ GOSUB

WIRKUNG : Diese Form von TIME$ erlaubt oder verbietet die Ausführung von ON TIME$ GOSUB Anweisungen.

HINWEISE : **TIME$ OFF**

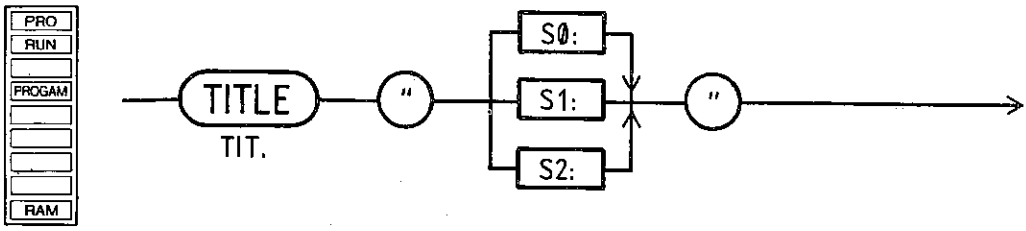
verhindert, daß mit der Anweisung ON TIME$ GOSUB in ein Unterprogramm verzweigt werden kann.

TIME$ ON

erlaubt die Annahme der Anweisung ON TIME$ GOSUB und damit zeitabhängige Verzweigungen.

TIME$ STOP

verhindert ebenso wie TIME$ OFF die Ausführung von ON TIME$ GOSUB Anweisungen. Ist jedoch die Verzweigungsbedingung erfüllt, das heißt die vereinbarte Zeit erreicht, wird diese Tatsache in einem Zwischenspeicher vermerkt und bei der nächsten TIME$ ON Anweisung die Verzweigung vorgenommen. STOP ist die standardmäßig angenommene Option für TIME$.



WIRKUNG : TITLE selektiert eines der ROM-Module oder den Arbeitsspeicher.

HINWEISE : In den auf der Rückseite des Computers befindlichen Modul-Fächern können insgesamt bis zu zwei RAM-Module installiert werden. Mit TITLE kann entschieden werden, welches Modul hiervon als Programmspeicher dienen soll oder ob hierfür wieder der interne Arbeitsspeicher des PC-1600 zu verwenden ist.

TITLE "S0:" selektiert den Arbeitsspeicher des PC-1600 als Programmspeicher. Dies ist auch die standardmäßige Annahme nach einem Total-Reset.

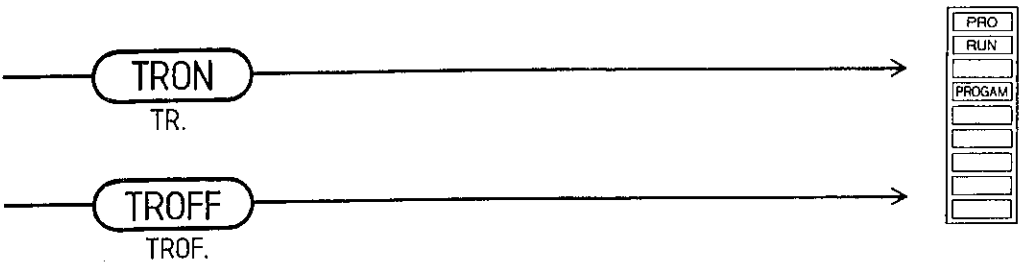
TITLE "S1:" selektiert das Modul des Faches S1

und

TITLE "S2:" das RAM-Modul des Faches S2 als Programmspeicher.

Ist das selektierte Modul als RAM-Disk initialisiert, wird ein ERROR-Code ausgegeben.

In einigen Situationen kann es nach Ausführung von TITLE vorkommen, daß während einer Programmunterbrechung die Tasten [↑] und [↓] ihre Wirkung zur Auflistung von Programmzeilen verlieren. Benutzen Sie in diesem Falle den LIST-Befehl oder lassen Sie das unterbrochene Programm einmal voll bis zum Ende durchlaufen, damit diese Tastenfunktion wieder hergestellt wird.



Siehe auch : CONT, STOP

WIRKUNG : Mit TRON und TROFF kann der TRACE-Modus ein- bzw. ausgeschaltet werden.

HINWEISE : Ist dieser Modus über den Befehl TRON aktiviert, hält der Computer nach Ausführung einer jeden BASIC-Zeile für die Dauer einer halben Sekunde an und gibt die Nummer der betreffenden Zeile auf der rechten Seite des Displays aus.

Wird während einer solchen Pause, in der die Anzeige der Zeilennummer erfolgt, die Taste gedrückt, so geht der Computer in den Einzelschritt-Modus über. Die Taste muß nun jedesmal erneut betätigt werden, damit die folgende Programmzeile zur Ausführung kommt. Wird diese Taste dauernd gedrückt, arbeitet der Computer die Programmzeilen nacheinander ab, ohne die zugehörigen Zeilennummern anzuzeigen. Eine gerade abgearbeitete Zeile kann durch Betätigung der Tasten und sichtbar gemacht werden. Mit der Taste läßt sich die Ausführung des Programmes dann nach Ablauf einer Wartezeit von 0.5 Sekunden weiter fortsetzen.

Stoppt das Programm infolge eines PRINT- oder INPUT-Befehles, kann es durch die Betätigung der ENTER-Taste weitergefahren werden.

Hält das Programm aufgrund eines STOP-Befehles an oder wurde es durch die BREAK-Taste abgebrochen, kann mit der CA Taste oder der Taste das Programm wieder in Betrieb genommen werden.

TROFF schaltet den Trace-Modus aus.

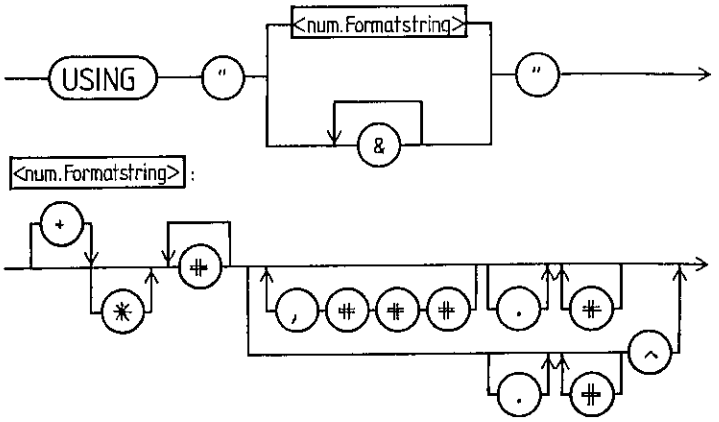
TRON und TROFF können auch programmiert werden. Der Trace-Modus bleibt dann solange bestehen, bis im Programm der nächste TROFF-Befehl vorgefunden wird.

BEISPIEL :

```
10:TRON
20:FOR I=1 TO 3
30:PRINT "I=",I
40:NEXT I

50:TROFF
60:END

>
RUN
                              10
                              20
I=                             1
                              30
                              40
I=                             2
                              30
                              40
I=                             3
>
```



Siehe auch : PAUSE, PRINT, PRINT#, LPRINT

WIRKUNG : USING erlaubt eine formatierte Datenausgabe.

HINWEISE : Das Format wird durch einen **<Format-String>** bestimmt, der sich aus folgenden Zeichen zusammensetzen kann:

- # Nummernzeichen
- & Kaufmannsund
- + Pluszeichen
- * Multiplikations-Operator
- ^ Potenz-Operator
- , Komma
- . Dezimalpunkt

Mit diesen Zeichen lassen sich so vielgestaltige Format-Strings bilden, daß es unmöglich ist,alle zulässigen Strings in einem Syntax-Diagramm zu erfassen. Aus diesem Grunde soll es hier lediglich als Richtschnur dienen, wie man die Zeichen zu sinnvollen Strings kombiniert. Darüberhinaus gibt es eine Vielzahl weiterer Kombinationsmöglichkeiten, die dieselbe Wirkung haben, aber teilweise unanschaulich sind.

USING kann als parametrische Anweisung innerhalb einer PAUSE-, PRINT- ,PRINT#- oder auch LPRINT-Anweisung verwendet werden.

Wird USING nicht als Parameter der genannten Befehle verwendet, sondern als alleinstehende Anweisung, gilt damit das damit eingestellte Format für alle folgenden PRINT- und LPRINT- und PAUSE-Anweisungen, bis es durch ein neues Format überschrieben wird. USING ohne Parameter, also vollkommen alleinstehend verwendet, hebt das Format auf.

Folgende acht grundlegende Formate lassen sich unterscheiden:

(1) "###" **Integer-Format:**

```
 43
-57
 -2
```

Bei drei Nummernzeichen lassen sich Integer-Werte mit bis zu zwei Stellen angezeigen. Ein Dezimalpunkt erscheint nicht.

(2) "###." **Fixkomma-Integer-Format:**

```
 98.
-43.
 -9.
```

In diesem Format werden alle Werte mit einem Dezimalpunkt versehen.

(3) "###.##" **Fixkomma-Format:**

```
 64.29
  5.00
-13.44
 -2.09
```

Es werden grundsätzlich zwei Stellen hinter dem Dezimalpunkt ausgegeben und vor diesem bis zwei Stellen dargestellt.

(4) "##.## " **Fließkomma-Format:**

```
 23.11E 05
-4.33E-02
 7.00E 00
```

In diesem Format wird der Wert einschließlich dem Exponentenzeichen E und einem unter Umständen vorzeichenbehafteten Exponenten angezeigt. Exponent, Exponentsymbol und Exponentvorzeichen belegen in jedem Falle vier Zeichen-Plätze.

(5) "###,###." **Integer-Tausender-Format**

```
 34,567.
      2.
   -230.
-2,345.
```

Es können fünf Ziffern einer Integer-Zahl angezeigt werden. Die Tausenderstellen werden dabei durch ein Komma abgetrennt.

(6) "+####" **Vorzeichenbehaftetes Format**

```
 -983
 +211
```

Es sind Integer-Werte mit bis zu drei Ziffern darstellbar. Jeder Wert wird mit seinem Vorzeichen versehen.

(7) "*####" Gefülltes Integer-Format

```
 2345      Es sind bis zu vier Ziffern
 **22      einer Integer-Zahl darstellbar.
 *230      Führende Lücken werden durch
-**66      einen Stern aufgefüllt.
```

(8) "&&&&&&" String-Format

```
ABCDEF     Es kann ein String mit bis zu
GHI        sechs Zeichen linksbündig ange-
           zeigt werden.
```

Die maximale Anzahl der Formatsymbole # oder * beträgt in den Formaten (1), (2), (3), (4), (6) und (7) 11 und im Format (5) 14.

Der PC-1600 gibt ohne Formatierungs-String die numerischen Daten mit bis zu 10 signifikanten Stellen aus.

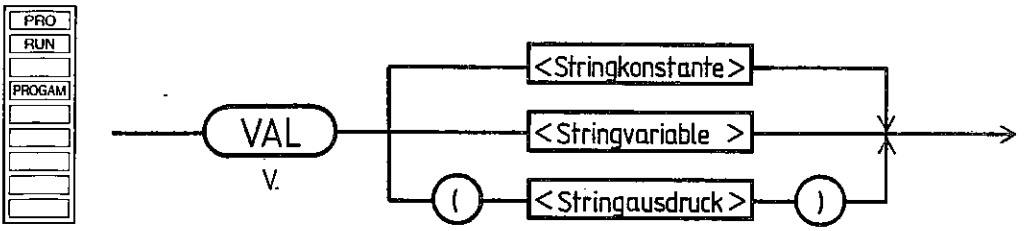
BEISPIELE :

```
10:H$="ABCDEF":WAIT 64
20:USING "&&":PRINT H$
30:PRINT USING "&&&&";H$
40:PRINT H$: USING:PRINT H$

>
RUN
AB
ABC
ABC
ABCDEF
>

10:Z=1234
20:PAUSE USING "#####";Z
30:PAUSE USING "#####.";Z
40:PAUSE USING "#####.##";Z
50:PAUSE USING "+####.##^";Z

>
RUN
                        1234
                       1234.
                     1234.00
                    +1.23E 03
```



Siehe auch : HEX$, STR$

WIRKUNG : VAL WANDELT einen String in einen numerischen Wert um.

HINWEISE : Die VAL-Funktion kann als Umkehrung der beiden Funktionen STR$ und HEX$ angesehen werden. Sie konvertiert einen String, der aus numerischen Zeichen besteht, in einen numerischen Wert.

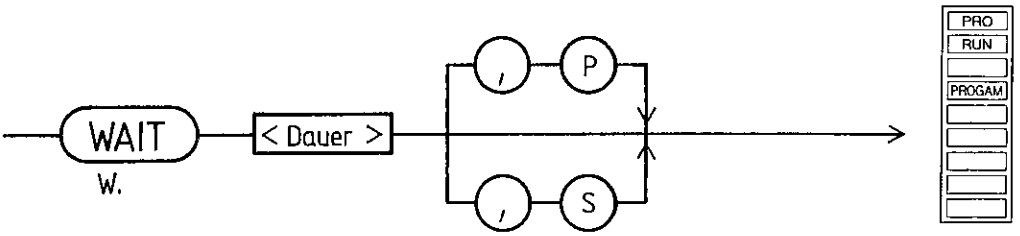
Ist das Funktionsargument ein dezimaler String, muß er aus den Zeichen 0 bis 9 zusammengesetzt sein. Er darf einen Dezimalpunkt und die Angabe eines Exponenten enthalten sowie ein Vorzeichen für die Mantisse und eines für den Exponenten aufweisen. In einem solchen Falle entspricht VAL genau der Umkehrung von STR$.

Ist das Funktionsargument dagegen ein hexadezimaler String, muß das erste Stringzeichen ein & sein und die nachfolgenden Zeichen aus solchen Symbolen bestehen, die zur Darstellung von Hex-Ziffern verwendet werden. In diesem Falle wirkt VAL als Umkehrung der HEX$-Funktion.

Enthält das Argument unzulässige Zeichen, wird der Funktionswert 0 geliefert.

BEISPIEL :

```
10:INPUT "FREQUENZ = ";A$
15:IF ASC(A$)<48 OR ASC(A$)>57 THEN 100
20:F=VAL(A$)
30:PRINT F
40:END
:
:
100:PRINT "NUR ZAHLENEINGABE ERLAUBT"
```



Siehe auch : PRINT

WIRKUNG : WAIT bestimmt die Zeitdauer, die nach Ausführung einer PRINT-Anweisung abzuwarten ist, bevor mit der nächsten Anweisung fortgefahren wird.

HINWEISE : WAIT ohne Parameterangabe setzt die Wartezeit auf unendlich. Nach einer PRINT-Anweisung stoppt das Programm also vollständig. Wie man aber aufgrund des angezeigten Bereitschaftszeichens > erkennen kann, ist der Programmablauf aber nicht abgebrochen. Er pausiert lediglich und kann mit Betätigung der ENTER-Taste fortgesetzt werden.

Wird der Parameter **<Dauer>** angegeben, bestimmt dieser die Wartezeit als ein Vielfaches einer 1/64 Sekunde. Der Wert 64 ergibt also eine Pause von 1 Sekunde, der Wert 128 eine von 2 Sekunden usw. Der erlaubte Wertebereich erstreckt sich von 0 bis 65535.

Zusätzlich kann eine der Optionen P oder S angegeben werden. Dieses ist jedoch ausschließlich im MODE 0 (PC-1600-Modus) möglich und auch nur dann, wenn der Parameter <Dauer> vorhanden ist.

Die Option P entspricht der standardmäßigen Einstellung und kann, wenn nicht Option S gewünscht ist, auch weggelassen werden. Sie bedeutet, daß die Pause unmittelbar nach jeder PRINT-Anweisung stattfindet.

Die Option S bestimmt, daß die Pause nur dann erfolgt, wenn das Display in allen vier Zeilen mit Daten gefüllt ist und die Absicht hat zu "Scrollen", d.h. die Inhalte der Zeilen um je eine Zeile nach oben versetzen möchte.

Wird ein Programm mit RUN gestartet, hängen die standardmäßigen WAIT-Einstellungen vom gültigen Anzeigemodus ab:

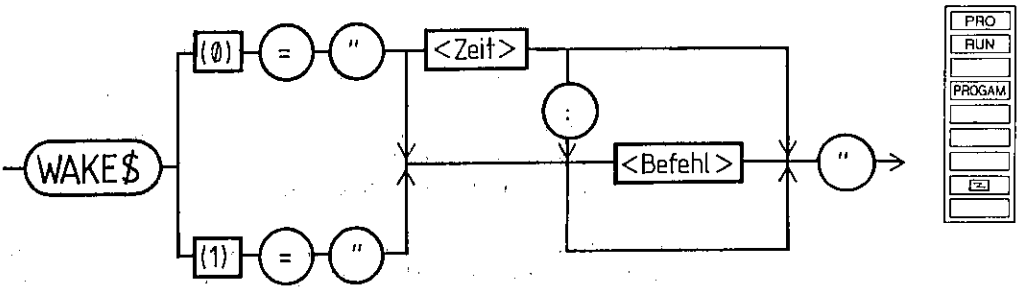
MODE 0 (PC-1600-Modus):	Keine Pause bei Ausführung einer PRINT-Anweisung.
MODE 1 (PC-1500-Modus):	Programmausführung pausiert für eine unbestimmte Zeit und muß durch Betätigung der ENTER -Taste fortgesetzt werden.

BEISPIEL :

```
10:FOR I=1 TO 10
20:WAIT (64*I)
30:PRINT "*";
40:NEXT I
50:WAIT
60:END

>
RUN
**********
>
```

Jeder Stern erscheint 1 Sekunde später als der vorhergehende.



Siehe auch : KBUFF$

WIRKUNG : WAKE$ bestimmt die Uhr-Zeit, bei der sich der ausgeschaltete Computer selbsttätig einschaltet und welche Befehl er dann abarbeiten soll.

HINWEISE :

WAKE$(0)="<Zeit>:<Befehl>"

bestimmt, zu welcher Zeit der Computer sich einschalten und welchen Befehl er anschließend ausführen soll. In Verbindung mit dem KBUFF$-Befehl erlaubt dieses die automatische Abarbeitung von "Batch-Dateien" zu einer vorgegebenen Zeit. Der Parameter <Zeit> muß dabei im Format:

MM/DD/HH/mm (Monat/Tag/Stunde/Minute) vorliegen.

WAKE$(0)=""

setzt WAKE(0)="<Zeit>:<Befehl>" außer Kraft.

WAKE$(1)="<Befehl>"

sorgt dafür, daß sich der Computer genau dann einschaltet und mit der Ausführung des Befehls-Strings beginnt, wenn am RS-232C-Interface das Signal CI (Pin 9) in den Zustand "high" geht. (Siehe auch Teil III, Kapitel 6)

WAKE$(1)=""

schaltet WAKE$(1)="<Befehl>" wieder ab.

Die Länge des Befehls-Strings kann maximal 26 Zeichen betragen. Soll der Befehl nicht nur in der Anzeige erscheinen, sondern auch tatsächlich ausgeführt werden, muß an den WAKE$-String ein "carriage return" angehängt werden. Dieses kann mit der CHR$-Funktion geschehen (s. Beispiel).

Ist der Computer bereits eingeschaltet, wenn ein WAKE$-Befehl aktiviert wird, ertönt ein Alarmsignal.

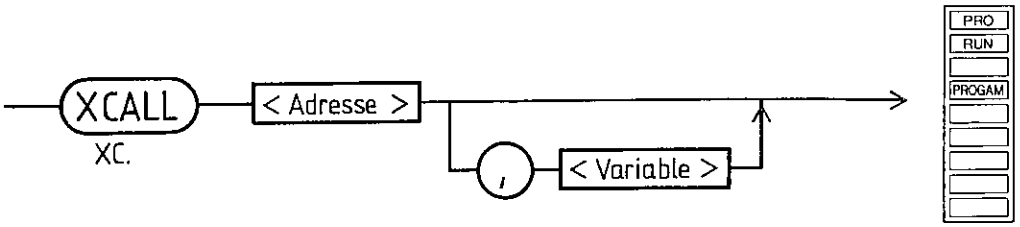
BEISPIEL : Wenn Sie das folgende Programm eingeben:

```
10:PRINT "Guten Morgen !"
20:PRINT "Schau doch einmal nach, ob"
30:PRINT "der Nikolaus schon da war."
40:END
```

und anschließend die Anweisung:

```
WAKE$(0)="12/06/07/30:RUN"+CHR$(&D)
```

erteilen und den Computer ausschalten, können Sie sich am Nikolaustag mit einer passenden Meldung wecken lassen.



Siehe auch : NEW, POKE, XPOKE

WIRKUNG : XCALL ruft ein Maschinensprache-Programm auf.

HINWEISE : XCALL und CALL sind grundsätzlich zwei gleiche Befehle. Im Gegensatz zum CALL-Befehl dient der Befehl XCALL jedoch zum Start von zum PC-1500 kompatiblen Maschinensprache-Programmen.
Es gilt also:

XCALL ruft eine PC-1500-kompatible Routine auf.
CALL ruft eine Z80-kompatible Routine auf.

Beachten Sie, daß der Befehl XCALL des PC-1600 dem Befehl CALL des PC-1500 entspricht.

Bei Aufruf der Maschinensprache-Routine kann ein einzelner Variablen-Wert übergeben werden. Nach Ablauf des Maschinensprache-Programmes wird dann der aktualisierte Wert an diese Variable zurückgegeben. Für die Wert-Übergabe stehen die beiden Prozessor-Register A und X bereit. Damit eine Wertübergabe erfolgen kann, muß die vereinbarte Variable natürlich auch wirklich existieren.

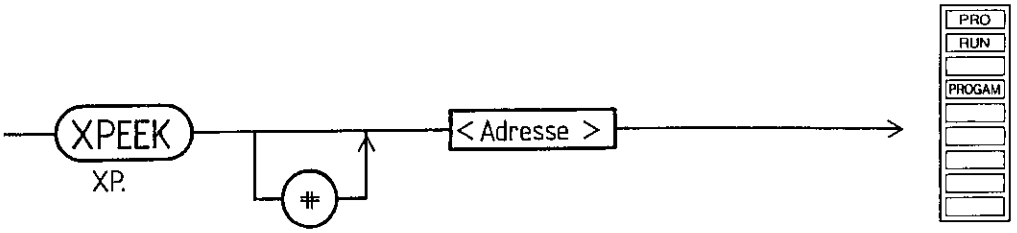
<Adresse> bestimmt die Adresse in der momentan gültigen Speicherbank, ab der das Programm zu starten ist.

<Variable> bestimmt die Variable, deren Wert an das Maschinensprache-Programm übergeben werden soll. Handelt es sich hierbei um eine numerische Variable, muß diese vom Typ Integer sein und einen Wert im Bereich von -32768 bis 32767 aufweisen. Ihr Wert wird dann an das X-Register abgegeben und nach Ablauf des Programmes der dann im Register stehende Wert der Variablen zugewiesen.

Bei einer String-Variablen wird an an Register X die Adresse übergeben, ab der der String abgelegt ist und in Register A die Länge des Strings eingetragen.

BEISPIEL :

```
:
:
400:XCALL 57405,X
410:PRINT "DER VOM MASCHINENSPRACHE-"
420:PRINT "PROGRAMM GELIEFERTE WERT"
430:PRINT "LAUTET : ";X
:
:
```



Siehe auch : PEEK, POKE, XPOKE

WIRKUNG : XPEEK liest ein Byte aus einer spezifizierten Speicherzelle.

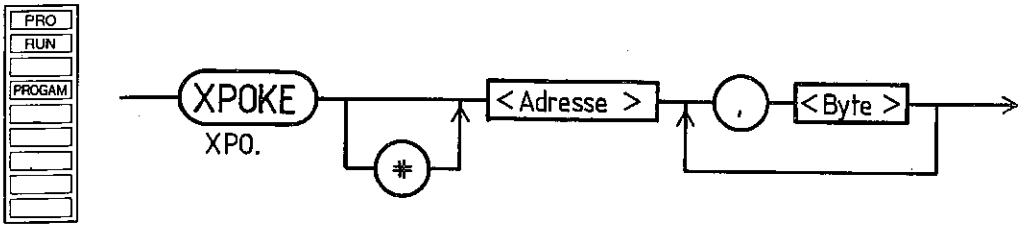
HINWEISE : XPEEK und PEEK sind von ihrer Wirkungsweise her grundsätzlich identisch. XPEEK hält sich jedoch an das beim PC-1500 übliche Format des dortigen PEEK-Befehles. XPEEK des PC-1600 ist also nichts weiter als ein Äquivalent des PC-1500-Befehles PEEK.

<Adresse> bestimmt eine Adresse im Speicherbereich 0 (ME 0).

#<Adresse> bestimmt eine Adresse im Speicherbereich 1 (ME 1). Hierbei gilt die momentan selektierte Speicherbank.

Weitere Einzelheiten hierzu können Sie aus der in Anhang D gezeigten Aufteilung des Speichers erkennen.

XPOKE



Siehe auch : PEEK, POKE, XPEEK

WIRKUNG : XPOKE schreibt ein Byte in die spezifizierte Speicheradresse oder mehrere Bytes mit dieser Adresse beginnend in nacheinander folgende Speicherzellen.

HINWEISE : Die Befehle XPOKE und POKE sind in ihrer Wirkung grundsätzlich gleich. XPOKE stellt jedoch ein Äquivalent des POKE-Befehles dar, wie er vom PC-1500 verstanden wird.

<Adresse> bestimmt die Adresse im Speicherbereich 0 (ME 0), in die das erste Byte der angegebenen Byte-Liste geschrieben werden soll.

#<Adresse> bestimmt die Adresse im Speicherbereich 1 (ME 1), in die das erste Byte der angegebenen Byte-Liste zu schreiben ist.

<Byte> bestimmt den jeweils gerade in die aktuelle Speicherzelle zu ladenden Wert zwischen 0 und 255.

Weitere Einzelheiten hierzu können Sie der im Anhang D beschriebenen Aufteilung des Speicherbereiches entnehmen.

TEIL V

ANHANG

A. Auswechseln der Batterien

B. Austausch der RAM-Module

C. Tabellen der Zeichen-Codes

D. Speicher-Aufteilung

E. Maschinensprache-Programme

F. ERROR-Codes des PC-1600

G. Liste der BASIC-Befehle

H. Kompatibilität zum PC-1500 und dessen Optionen

I. Vorsichtsmaßnahmen und Problemlösungshinweise

J. Technische Spezifikationen

K. Syntax-Diagramme

Auswechseln der Batterien

Der PC-1600 meldet eine ungenügende Stromversorgung durch zu schwach gewordene Batterien oder Akkus auf zweierlei Art:

- Sind die Batterien des PC-1600 oder aber die Akkus des angeschlossenen Druckers (CE-1600P, CE-150) zu schwach, erscheint in der Statuszeile des Displays das Symbol:

  BATT

- Erweisen sich die Batterien bzw. Akkus während einer Programmausführung als zu schwach, wird das gerade laufende Programm abgebrochen und ein entsprechender ERROR-Code ausgegeben. Anhand dieses Codes können Sie erkennen, ob die Fehlerursache in den Batterien des Computers oder aber in den Akkus des angeschlossenen Peripheriegerätes begründet liegt. Die Bedeutung des ERROR-Codes kann dem Anhang F entnommen werden.

HINWEIS : Beachten Sie, daß bei ausgeschaltetem Computer die Pufferung der im Arbeitsspeicher enthaltenen Daten und Programme nur solange erfolgen kann, wie die eingesetzten Batterien in Ordnung sind. Bei zu schwachen, entladenen oder nicht eingesetzten Batterien kann diese Speicherpufferung (memory save guard) nicht funktionieren. Dieses betrifft auch die RAM-Module, die keine eingebaute Batterie besitzen.

Sofern sich im Arbeitsspeicher keine wichtigen Daten befinden, können Sie einen Batteriewechsel so vornehmen, als ob Sie bei einem neuen Gerät zum ersten Male die Batterien einsetzten. Folgen Sie hierzu den in Abschnitt 2.1 gegebenen Installationsbeschreibungen und lösen Sie anschließend einen Total-Reset aus.

Falls jedoch Daten und Programme im Speicher vorliegen, die Sie nicht gerne verlieren möchten, so gehen Sie bitte nach einem der beiden nachstehend genannten Wege vor:

WEG 1 : Schließen Sie einen Netzadapter an den Computer an, um dessen Stromversorgung während des Batterieaustausches sicherzustellen. Halten Sie sich dann an die folgenden fünf Schritte:

1) Schalten Sie den Computer aus.

2) Schließen Sie den Netzadapter an, wobei zuerst das Netzkabel in die Steckdose und erst dann die Verbindung zum PC-1600 herzustellen ist.

3) Wechseln Sie die Batterien.

4) Entfernen Sie den Netzadapter in umgekehrter Weise wie Sie ihn zuvor angeschlossen haben.

5) Schalten Sie den Computer an.

Daten und Programme sollten bei Einhaltung dieser fünf Schritte erhalten geblieben sein. Ein Reset ist nicht erforderlich (und auch nicht ratsam).

WEG 2 : Der zweite Weg ist etwas komplizierter. Hierbei sind alle wichtigen Daten und Programme auf einer Diskette, einer Cassette oder einem Modul (Programm-Modul oder RAM-Disk) vor dem Batteriewechsel zu sichern. Nach dem Batterieaustausch können die gesicherten Programme und Daten wieder zurückgeladen werden. Für die Durchführung der gesamten Prozedur sollten Sie sich an die folgenden Schritte halten:

1) Sofern keine Option zur Speicherung angeschlossen ist, schalten Sie den Computer aus.

2) Schließen Sie nun ein Diskettenlaufwerk oder einen Cassetten-Recorder an oder setzen sie ein Programm-Modul oder aber ein als RAM-Disk fungierendes Modul in den Computer ein.

3) Schalten Sie den Computer ein und schreiben Sie die gewünschten Daten und Programme mit Hilfe des SAVE-Befehles auf das gewählte Speichermedium.

4) Schalten Sie den Computer aus und wechseln Sie die Batterien.

5) Initialisieren Sie den Computer. Beachten Sie dazu die Beschreibungen des Abschnittes 3.3.

6) Benutzen Sie den LOAD-Befehl, um die Programme oder Daten in den Computer zurückzuholen.

Austausch der RAM-Module

Es können zwei RAM-Modul-Arten unterschieden werden:

PROGRAMM-MODULE: besitzen eine eingebaute Batterie und können deshalb die in ihnen befindlichen Daten auch außerhalb des Computers halten, d.h. puffern. Sie eignen sich damit in idealer Weise zur Speicherung von Programmen, die man bei Bedarf schnell laden möchte. Ihre Handhabung ist besonders einfach und problemlos.

SPEICHER-MODULE: besitzen keine eingebaute Batterie. Somit sind sie nicht in der Lage, Programme außerhalb des Computers zu puffern. Sie eignen sich deshalb lediglich für eine Erweiterung des PC-1600-internen Arbeitsspeichers. Nachdem ein Speicher-Modul gerade installiert und der Computer eingeschaltet worden ist, erscheint die Meldung: NEW0?:CHECK auf dem Display. Der Computer hat also eine Veränderung des Speichers bemerkt. Er bittet Sie mit dieser Meldung für eine Bereinigung des Speichers zu sorgen. Dazu sind folgende Schritte durchzuführen:

a) Mit der [CL]-Taste die Meldung quittieren.
b) Falls erforderlich, mit der MODE -Taste den Computer in den PRO-Modus stellen.
c) Durch Eingabe des Befehles NEW0 (und anschließender Betätigung der [ENTER] -Taste) den Speicher löschen und für BASIC-Programme reservieren.

Die besagte Meldung erscheint nicht, wenn ein Programm-Modul in den Computer eingesetzt wird. Eine Löschung mit dem Befehl NEW0 ist nicht erforderlich. Sofern die Programm-Module nicht mit dem TITLE-Befehl selektiert werden, gelten sie als unabhängig vom Anwender-Bereich.

Die besagte Meldung erscheint nicht, noch ist eine Löschung mit NEW0 erforderlich, wenn ein Programm-Modul eingesetzt wird. Ein Programm-Modul ist unabhängig vom Anwender-Bereich, solange es nicht mit dem TITLE-Kommando selektiert worden ist.

Bei einer Speichererweiterung ist zuerst immer das Fach S1 zu belegen, bevor auch Fach S2 benutzt werden kann !

Mit dem MEM-Befehl können Sie sich nach der Installierung eines Modules vergewissern, ob der zusätzliche Speicherbereich angenommen worden ist. Ohne Modul stehen dem Anwender maximal 12090 freie Speicherplätze zur Verfügung.

Tabellen der Zeichen-Codes

Zeichen-Codes im MODE 0

Im MODE 0 schließt der Zeichensatz des PC-1600 internationale Sonderzeichen, grafische Symbole und griechische Schriftzeichen ein, die zusätzlich zu den üblichen Buchstaben, Ziffern, Satzzeichen und sonstigen ASCII-Symbolen dargestellt werden können. Dieser Zeichensatz entspricht dabei genau dem eines IBM-PC.

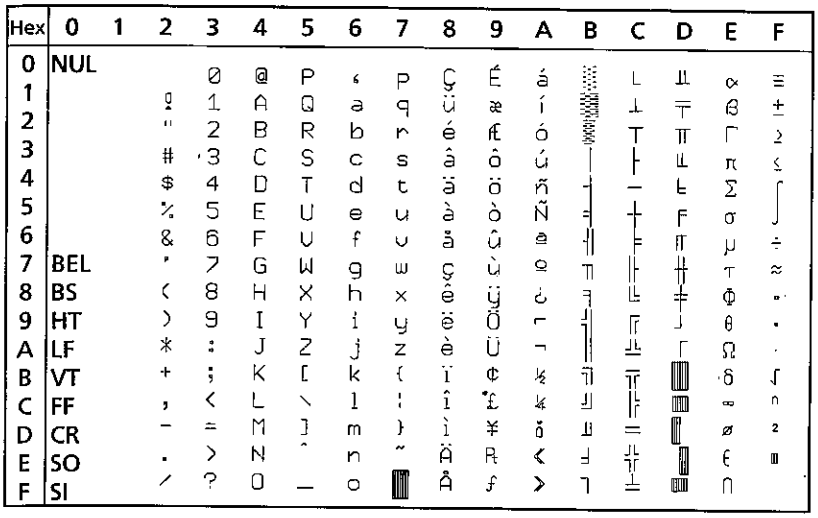


Abbildung 51 : Zeichen-Codes des PC-1600

Um aus dieser Tabelle den hexadezimalen Code für ein bestimmtes Zeichen zu entnehmen, suchen Sie dieses in der Tabelle auf. Die Spalte, in der sich das Zeichen befindet, beschreibt die höherwertige und die Zeile die niederwertige hexadezimale Ziffer. Der hexadezimale Code (oder einfach: HEX-Code) für den kleinen Buchstaben "a" lautet daher &61.

Jedes in dieser Tabelle enthaltene Zeichen kann auf dem Display mit Hilfe der BASIC-Funktion CHR$ angezeigt werden.

Bei vertikaler Druckrichtung (s. ROTATE-Befehl) werden diese Zeichen um 90° gedreht dargestellt.

Zeichen-Codes im MODE 1

Im MODE 1 wird der Zeichensatz des PC-1600 modifiziert, damit er Kompatibilität zum PC-1500 gewähren kann. Die folgende Tabelle zeigt, welche Unterschiede in den beiden Anzeige-Modi bestehen:

HEX CODE	27	5B	5C	5D	5E	5F	60	7B	7C	7D	7E	7F
MODE Ø	'	[	¥	]	^	_	`	{	|	}	~	
MODE 1		⌠	¥	∏	√						~	

Abbildung 52

Speicher-Aufteilung

Der PC-1600 kann 8 Speicherbereiche (memory banks) adressieren. Es sind dies die Bereiche "bank 0" bis "bank 7". Die ersten vier sind für RAM-Bausteine (random access memory) gedacht. Die vier weiteren enthalten die System-ROM-Bausteine (read only memory) und dienen desweiteren der Adressierung von Peripheriegeräten. Bank 7 ist zwar adressierbar, aber ungenutzt.

Internes RAM (vom Z80-Hauptprozessor aus gesehen):

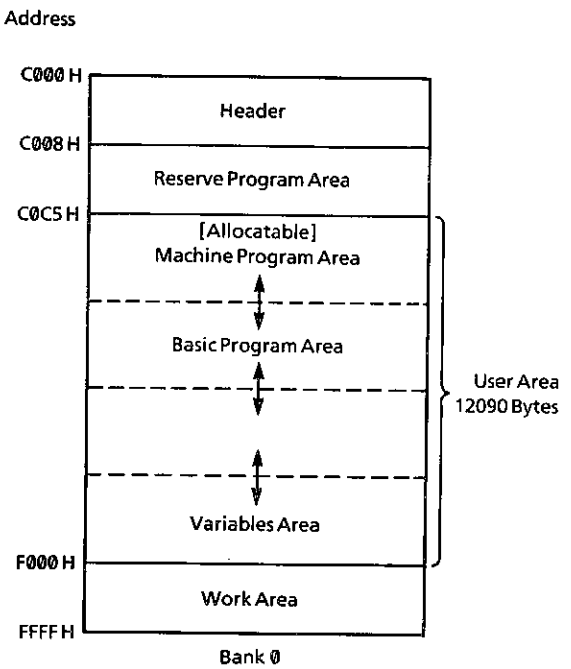


Abbildung 53 : Internes RAM

Ohne Module stehen dem Anwender vom RAM des PC-1600 für eigene Programm-Zwecke maximal 12090 Bytes zur Verfügung. Die obige Abbildung zeigt, wie ein Teil dieses Benutzerbereiches für die Aufnahme von Maschinensprache-Programmen abgezweigt werden kann. Dieser Bereich für die Maschinensprache-Programme kann mit dem Befehl NEW zu Null gesetzt werden.

Internes RAM und Erweiterungsbereiche (bank 0 bis bank 3) vom Hauptprozessor aus gesehen:

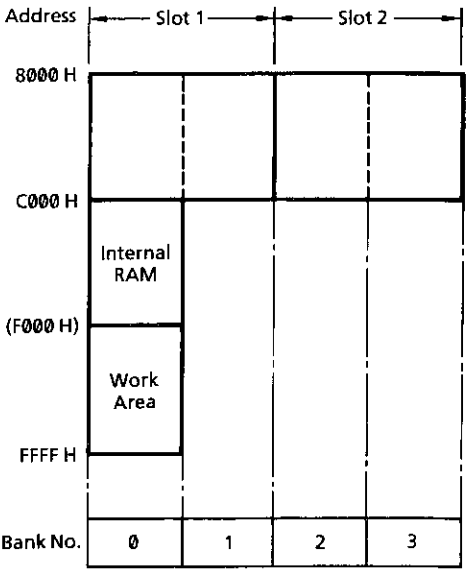


Abbildung 54 : Internes RAM und Modulfächer

Gesamtspeicher-Aufteilung (vom Hauptprozessor Z80 aus gesehen):

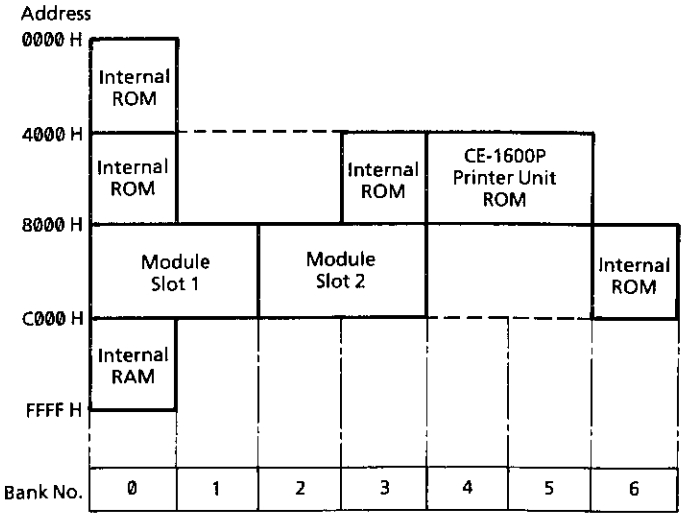


Abbildung 55 : Gesamtspeicher

Gesamtspeicher-Aufteilung (gesehen vom Nebenprozessor LH-5803):

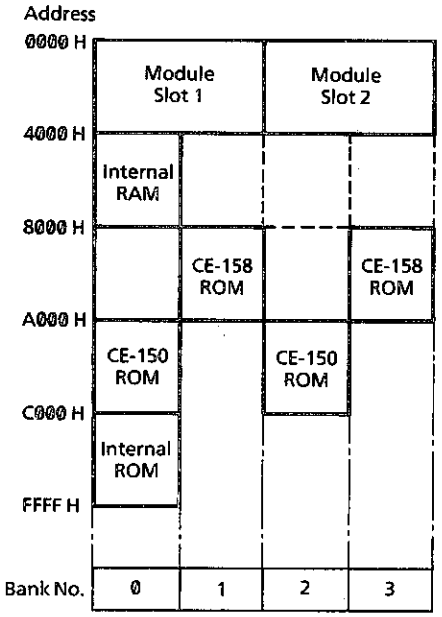


Abbildung 56 : Gesamtspeicher

Maschinensprache-Programme

Der PC-1600 kann von fortgeschrittenen und erfahrenen Anwendern direkt in Z80-Maschinen-Code programmiert werden.

BASIC-Befehle zur Maschinensprache-Programmierung

Das PC-1600-BASIC stellt einige Befehle bereit, mit denen sich Maschinensprache-Programme erstellen, sichern und laden lassen. Weitere Befehle erlauben die Ausführung dieser Programme und die direkte Steuerung der Ein-/Ausgabe-Schnittstellen (I/O ports). Manche Befehle adressieren dabei den zum Z80 kompatiblen Hauptprozessor, andere dagegen Nebenprozessor LH-5803. Die besagten Befehle für diese Maschinensprache-Ebene lauten:

BLOAD, BSAVE, CALL, CLOAD M, CSAVE M, INP, OUT, PEEK, POKE, XCALL, XPEEK, XPEEK#, XPOKE und XPOKE#.

Speicherplatzreservierung für Maschinensprache-Programme

Das interne RAM kann mit dem NEW-Befehl für die Aufnahme von Maschinensprache-Programmen vorbereitet werden. Dabei wird die niedrigste Speicheradresse für die Ablage von BASIC-Programmen bestimmt. Der Anwender kann mittels Maschinensprache auch auf andere Speicherbereiche und Systemroutinen zugreifen. Dieses erfordert aber eine genaueste Kenntnis der gesamten Speicheraufteilung des PC-1600. (Siehe hierzu Anhang D.)

Kompatibilität mit dem PC-1500

Folgende Tabelle beschreibt die maschinensprache-bezogenen BASIC-Befehle, wie sie im PC-1500 und PC-1600 vorzufinden sind, und welche davon kompatibel sind:

PC-1600-Befehl	PC-1500-Befehl	Funktion
XCALL	CALL	Startet ein Maschinenprogramm, das für den Nebenprozessor LH-5801/3 codiert ist.
CALL		Startet ein Maschinenprogramm, das für den Z80-kompatiblen Hauptprozessor codiert ist.
XPOKE	POKE	Schreibt ein Byte in eine vom LH-5801 verwaltete Adresse.
POKE		Schreibt ein Byte in eine vom Z80 verwaltete Adresse.
XPEEK	PEEK	Liest ein Byte aus einer vom LH-5801 verwalteten Adresse.
PEEK		Liest ein Byte aus einer vom Z80 verwalteten Adresse.
XPOKE#	POKE#	Sendet ein Byte zu einem vom LH-5801/3 verwalteten Port.
XPEEK#	PEEK#	Liest ein Byte von einem vom Z80 verwalteten Port.

ERROR-Codes des PC-1600

Programm-Fehler

Code Beschreibung des Programm-Fehlers

1 Syntax-Fehler.
Die Anweisung entspricht nicht den syntaktischen Regeln des SHARP-BASICs. Es liegt ein falscher Betriebs-Modus vor: Versuch ein Programm im PRO-Modus zu starten oder im RUN-Modus zu programmieren.

2 NEXT-Anweisung ohne FOR.
Das Programm enthält einen NEXT-Befehl ohne zugehörige FOR-Anweisung. Entweder fehlt diese Anweisung wirklich oder enthält irrtümlicherweise eine andere Laufvariable.

Beispiel:
```
FOR I=1 TO 20
....
....
NEXT J
```

Der Fehler kann genausogut auch an einer unsauberen Ver-Schachtelung von FOR/NEXT-Schleifen liegen, wie z.B. im Falle:

```
FOR I=1 TO 20
FOR J=1          TO 10
...
...
NEXT I
NEXT J
```

Ebenso ist es möglich, daß eine NEXT-Anweisung durch einen fehlgesteuerten Programmablauf (z.B. einer unerlaubten Sprunganweisung) auf eine bereits abgearbeitete NEXT-Anweisung trifft.

4 READ ohne DATA-Anweisung.
Es liegt ein READ-Befehl ohne zugehörige DATA-Anweisung vor. Möglich ist auch, daß in der DATA-Zeile zu wenig Daten enthalten sind bzw. die READ-Anweisung mehr als erforderlich aufgerufen wird.

Beispiel: Es sollen sieben Daten gelesen werden, obwohl nur fünf zur Verfügung stehen:

```
10 FOR X=1 TO 7
20 READ N(X)
30 NEXT X
:
:
80 DATA 34,765,22,88,17
90 END
```

Code	Beschreibung des Programm-Fehlers

5 Mehrfach-Deklaration.
Versuch, eine existierende Array-Variable ein zweites Mal zu deklarieren. Vor einer neuen DIMensionierung muß das Array erst gelöscht werden.

6 Undeklariertes ARRAY.
Die vorgefundene Array-Variable existiert nicht, d.h. sie wurde nicht mit der DIM-Anweisung deklariert.

7 Typen-Verwechslung.
Variable und Datum passen typenmäßig nicht zueinander. Es wird z.B. versucht, einer numerischen Variablen einen String zuzuweisen oder umgekehrt einer Stringvariablen ein numerischer Wert.

Beispiele: LET A$=234, LET A="ALPHA", B=SIN(A$)

8 Versuch, ein Array mit mehr als zwei Dimensionen zu deklarieren.

Beispiel: DIM XM(4,5,2)

9 Ein Index einer Array-Variablen liegt außerhalb des als zulässig vereinbarten Bereiches.

Beispiel:
```
10 DIM P(2,3)
...
50 INPUT P(2,4)
```

10 Speicherbereich zur Aufnahme von Variablen ist voll.

11 Nichtexistierende Zeile.
Die in einer Verzweigungsanweisung (z.B. GOTO, GOSUB) angegebene Zeile ist im Programm nicht enthalten.

Code	Beschreibung des Programm-Fehlers
12	Unzulässiger Format-String in der USING-Anweisung.
13	Arbeitsspeicher ist voll Der Speicherplatz reicht nicht mehr aus, um weitere Programmzeilen oder Variablen aufzunehmen.
14	Stack voll Die Verschachtelung von FOR/NEXT-Schleifen ist zu tief oder die Berechnung von verschachtelten Funktionen zu komplex.
15	Unterprogramm-Stack ist voll Es sind zuviele Unterprogramm-Aufrufe ineinandergeschachtelt oder es liegt ein String mit einer Länge von mehr als 80 Zeichen vor.
16	Exponent gefunden, dessen Betrag größer als 99 lautet.
17	Unerlaubte Daten-Typen innerhalb eines arithmetischen Ausdrucks. Es wird versucht, arithmetische Operationen auf Stringvariable auszuüben. Beispiele: A$*B$, C$/D$
18	Unerlaubte Parameter-Spezifikation einer Funktion.
19	Parameter liegt außerhalb des zulässigen Bereiches.
20	Unerlaubte Spezifikation einer Standard-Variablen.
21	Verwendung eines numerischen Ausdruckes anstelle einer numerischen Variablen.
22	Speicher voll. Es kann kein Programm geladen werden.
23	Unerlaubte Spezifikation des TIME$-Strings.
24	Versuch, ein mit einem Kennwort geschütztes Programm zu listen oder zu editieren.
25	Unzulässige Adreßangabe innerhalb einer NEW-Anweisung.
26	Unzulässiger Betriebsmodus: Der angegebene Befehl kann in der momentanen Betriebsart nicht ausgeführt werden.
27	Unbekannter Befehl. Die durch den Befehl angesprochene Peripherie oder Option ist nicht angeschlossen.

Code	Beschreibung des Programm-Fehlers
28	Bei der Durchführung der INPUT- bzw. AREAD-Anweisung ist ein unzulässiger String entdeckt worden, der einem BASIC-Befehlswort entspricht.
30	Zeilen-Nummer größer als 65539.
32	Die mit GCURSOR eingestellte Cursor-Position verhindert die Anzeige der durch einen INPUT-Befehl angeforderten Daten.
35	PC-1500-Peripherie nicht angeschlossen.
36	Die Daten können nicht in dem durch die USING-Anweisung vorgegebenen Format angezeigt werden.
38	Division durch Null. Der sich aus einem Ausdruck errechnende Divisor ist zu Null geworden oder aber die als Divisor eingesetzte Variable ist noch nicht definiert worden.
39	Unzulässiger Funktionsaufruf. Das Argument einer SQR- oder LN-Funktion ist negativ. Anstelle eines Integer-Argumentes enthält die Funktion einen Wert mit Nachkommastellen.
100	Umnumerierung der Programmzeilen mittels RENUM-Kommando ist nicht erfolgreich abgeschlossen.
101	Unerlaubte Geräte-Spezifikation innerhalb einer TITLE- oder NEW-Anweisung.
102	Ungültige Geräte-Selektion. (Gerät nicht angeschlossen)
103	RAM-Modul voll. Die INIT-Parameter können nicht gesetzt werden.
104	Die ON..GOSUB-Anweisung enthält mehr als acht Interrupt-Verzweigungen.
105	Zu viele Software-Interrupt-Ebenen
106	RETI-Anweisung ohne dazugehörige ON....GOSUB-Anweisung vorgefunden.
107	RESUME-Anweisung ohne dazugehörige ON...ERROR...GOSUB-Anweisung vorgefunden.
108	Kennwort kann nicht gelöscht werden.
109	Unzulässiger Variablen-Name in der ERASE-Anweisung.

Code	Beschreibung des Programm-Fehlers
110	Der PC-1500-Modus (MODE 1) kann nicht gesetzt werden. Befehl im Betriebsmodus MODE 0 nicht zulässig. Die auf die PC-1500-Peripherie bezogenen Befehle arbeiten nur im MODE 1.
111	Unzulässiger Modus.
112	Zeile zu lang. Während des Editierens eines Programmes überschritt die Zeile die Länge von 80 Zeichen.
131	String-Wert mit Vorzeichen versehen.

Recorder-Fehler

Code	Beschreibung des Recorder-Fehlers
40	Syntax-Fehler innerhalb eines Recorder-Befehles.
42	Nicht genügend Speicherplatz vorhanden, um ein Programm von Cassette laden zu können.
43	Verifikations-Fehler. Die CLOAD?-Anweisung stellt fest, daß das im Speicher befindliche Programm nicht mit dem auf der Cassette befindlichen Programm übereinstimmt.
44	Lese-Fehler. Inkompatible Aufzeichnung oder fehlerhafte Einstellung von Lautstärke- und Tonhöhe.

Drucker-Fehler

Code	Beschreibung des Drucker-Fehlers
70	Stift befindet sich außerhalb der X- oder Y-Koordinaten des zulässigen Bereiches von -2048 bis 2047.
71	Die im TEXT-Modus vorgenommene Formular-Rückführung (umgekehrter Vorschub) beträgt mehr als 10,24 cm. (Betrifft nur die Option CE-150.)
72	Falsche Spezifikation der TAB-Parameter.
73	Unzulässiger Befehl im derzeit gültigen Modus: Benutzung von Grafik-Befehlen im Text-Modus oder umgekehrt.
74	Zu viele Parameter in der LINE- oder RLINE-Anweisung

Code	Beschreibung des Drucker-Fehlers
76	Datenzeile ist länger als die mit PCONSOLE eingestellte Zeilenlänge. Ein numerisches Resultat paßt von der Länge her nicht in eine Zeile, um im Text-Modus dargestellt werden zu können.
78	LPRINT- oder LINE-Kommando kann nicht ausgeführt werden. Der Stift befindet sich am falschen Platz oder es liegt eine Blockierung des Druckers aufgrund unzureichend geladener Akkus vor.
79	Farbwechselsignal nicht an den Drucker weitergeleitet. (Betrifft nur die Option CE-150.)
80	Entladene Akkus. Drucker hat sich "aufgehängt".

Port-Fehler

Code	Beschreibung des Port-Fehlers
140	Unerlaubte Parameter-Angabe in der SET-Anweisung.
141	Die mit dem INIT-Befehl zu spezifizierende Größe des Empfangspuffers überschreitet entweder den Maximalwert von 16383 Bytes oder den derzeit verfügbaren Speicherplatz.
142	Empfangs-Fehler. Parität oder Einrahmung des Datenbytes fehlerhaft oder Überlauf des Empfangs-Puffers.
143	Wartezeit überschritten. Innerhalb der mit RCVSTAT festgelegten Zeit sind keine Daten von der Peripherie her empfangen worden.
144	Der in der SETDEV-Anweisung angegebene Port ist bereits offen.

Datei-Fehler

Code	Beschreibung des Datei-Fehlers
150	Zu viele Dateien in der MAXFILES-Anweisung spezifiziert.
151	Datei existiert bereits. Wählen Sie einen anderen Namen.

Code	Beschreibung des Datei-Fehlers
152	Datei nicht gefunden. Überprüfen Sie, ob die Angabe des Dateibezeichners korrekt ist. Beim Speichern mittels SAVE erweist sich die Diskette als voll.
153	Unzulässige Dateinummer-Angabe innerhalb einer Lese- oder Schreibanweisung. Die Datei mit der entsprechenden Nummer ist noch nicht für den Lese- oder Schreibzugriff geöffnet worden.
154	Die Datei ist bereits geöffnet. Schließen Sie sie, bevor Sie sie im richtigen Modus für den gewünschten Zugriff öffnen.
155	Unzulässige Laufwerksbezeichnung: Diskettenlaufwerk ist nicht angeschlossen.
156	Falsche Parameterangaben in der SET-Anweisung.
157	Unzulässiger oder fehlerhafter Dateiname.
158	Befehl im Zusammenhang mit dem Diskettenlaufwerk nicht anwendbar.
159	Versuch, auf eine geschützte Diskette zu schreiben.
160	Keine Diskette im Laufwerk
161	Diskette ist noch nicht mit INIT formatiert worden.
162	Schreib- oder Lesefehler bezüglich eingelegter Diskette.
163	Falsche Diskette im Laufwerk. Es fand ein Diskettenwechsel statt, obwohl noch eine geöffnete Datei vorlag.
164	Diskette voll.
165	Dateiende beim Lesen mit der INPUT# überschritten. Es wurden bereits alle Datensätze gelesen.
166	Nicht genügend Speicherplatz im Systembereich vorhanden, um das Diskettenlaufwerk zu bedienen.
167	Fataler Disketten-Fehler. Die Disketten-Inhalte oder die Diskette selbst sind zertört.
168	Diskettenlaufwerk funktioniert nicht korrekt. Es liegen zu schwache Akkus oder ein Hardware-Fehler vor.

Liste der BASIC-Befehle

Befehl	Kurzbeschreibung
ABS	Bildet den Absolutbetrag einer Zahl.
ACS	Arcus-Cosinus-Funktion.
ADIN ON/OFF/STOP	Erlaubt/verhindert analoge Interrupts.
AIN	Liefert Dezimalwert des analogen Pegels.
ALARM$	Bestimmt Alarm-Zeit und Alarm-Meldung.
AREAD	Liest angezeigten Wert in eine Variable.
ARUN	Startet ein BASIC-Programm automatisch.
ASC	Liefert den ASCII-Code eines Zeichens.
ASN	Arcus-Sinus-Funktion.
ATN	Arcus-Tanges-Funktion.
AUTO	Generiert Zeilennummern.
BEEP	Erzeugt verschiedene Töne.
BEEP ON/OFF/STOP	Schaltet den Tongeber ein bzw. ab.
BLOAD	Lädt ein Maschinen-Programm von Diskette.
BREAK ON/OFF/STOP	Schaltet Wirkung der BREAK-Taste ein/aus.
BSAVE	Sichert Maschinen-Programme auf Diskette.
CALL	Startet ein Maschinen-Programm.
CHAIN	Lädt von einem laufenden BASIC-Programm aus ein weiteres BASIC-Programm und startet es.
CHR$	Liefert das Zeichen eines ASCII-Codes.
CLEAR	Löscht alle Variablen.
CLOAD	Lädt ein BASIC-Programm von Cassette.
CLOAD ?	Überprüft ein geladenes BASIC-Programm.
CLOAD M	Lädt ein Maschinen-Programm von Cassette.
CLOSE	Schließt eine Datei.
CLS	Löscht das Display.
COLOR	Wählt den Farbstift des Druckers aus.
COM$	Liefert einen String mit den eingestellten Kommunikationsparametern.
COMn ON/OFF/STOP	Erlaubt bzw. verhindert Kommunikations-Interrupts.
CONT	Setzt ein mit STOP/BREAK unterbrochenes BASIC-Programm fort.
COPY	Kopiert Disketten-Dateien.
COS	Liefert den Cosinus eines Winkels.
CSAVE	Sichert BASIC-Programme auf Cassette.
CSAVEM	Sichert Maschinen-Programme auf Cassette.
CSIZE	Bestimmt die Größe der Druck-Zeichen.
CURSOR	Positioniert den Cursor.
DATA	Hält die mit READ zu lesenden Daten bereit.
DATE$	Enthält das aktuelle Datum.
DEG	Sexagesimal/Dezimal-Wandlung eines Winkels.
DEGREE	Schaltet Winkelmodus auf Altgrad.
DELETE	Löscht Programmzeilen.

Befehl	Kurzbeschreibung
DIM	Reserviert Platz für Variablen und Arrays.
DMS	Dezimal/Sexagesimal-Wandlung von Winkeln.
DSKF	Zeigt die freie Diskettenkapazität an.
END	Beendet ein BASIC-Programm.
EOF	Weist auf das Ende einer Datei hin.
ERASE	Löscht bestimmte Variablen und Arrays.
ERL	Liefert die Nummer der fehlerhafteten Zeile.
ERN	Liefert den Code eines anstehenden Fehlers.
EXP	Exponential-Funktion.
FILES	Gibt Inhaltsverzeichnis der Diskette aus.
FOR..NEXT	Erlaubt determinierte Programmschleifen.
GCURSOR	Positioniert den Grafik-Cursor.
GLCURSOR	Positioniert den Druckstift im Grafik-Modus.
GOSUB..RETURN	Führt Unterprogramme aus.
GOTO	Verzweigt zu einer bestimmten Zeile.
GPRINT	Zeichnet Grafik-Muster auf dem Display.
GRAD	Schaltet Winkelmodus auf Neugrad.
GRAPH	Setzt den Drucker in den Grafik-Modus.
HEX$	Liefert den Hex-String einer Zahl.
IF..THEN	Ermöglicht bedingte Programmverzweigungen.
INIT	Initialisiert Module, Disketten und den Empfangspuffer
INKEY$	Holt Zeichen aus dem Tastatur-Puffer.
INP	Liefert Daten vom Prozessor-Port.
INPUT	Erlaubt die Dateneingabe während eines laufenden BASIC-Programmes.
INPUT#	Liest Daten aus einer Datei.
INSTAT	Liefert den Zustand der Steuersignale eines Interfaces.
INSTR	Sucht einen Teilstring innerhalb eines vorgegebenen Strings.
INT	Liefert den ganzen Anteil einer Zahl.
KBUFF$	Schreibt Zeichen in den Tastatur-Puffer.
KEY ON/OFF/STOP	Schaltet Funktionstasten ein/aus.
KEYSTAT	Setzt/Löscht Wiederhol/Klick-Funktionen der Tasten.
KILL	Löscht eine Disketten-Datei.
LCURSOR	Bewegt Druckerstift zu bestimmter Position.
LEFT$	Liefert linksbündigen Teil-Strings.
LEN	Ermittelt die Länge eines Strings.
LET	Weist Variablen Werte zu.
LF	Sorgt für den Papiervorschub.
LFILES	Druckt ein Disketten-Inhaltsverzeichnis.
LINE	Verbindet Display-Punkte mit einer Linie.
LIST	Listet Programmzeilen in der Anzeige auf.
LLINE	Druckt die Verbindungslinie zweier Punkte.
LLIST	Druckt Programmzeilen.
LN	Liefert den natürlichen Logarithmus.
LOAD	Lädt eine Datei von Diskette.

Befehl	Kurzbeschreibung
LOC	Liefert Record-Nummer einer Datei.
LOCK / UNLOCK	Schaltet MODE-Taste ein/aus.
LOF	Liefert die Dateigröße.
LOG	Liefert den dekadischen Logarithmus.
LPRINT	Sendet Daten zum Drucker oder einem seriellen Interface.
MAXFILES	Bestimmt die Maximal-Anzahl offener Dateien.
MEM	Liefert Anzahl freier Speicherplätze.
MERGE	Fügt zwei Programme im Speicher aneinander.
MID$	Liefert den Mittelteil eines Strings.
MOD	Liefert den Rest einer Integer-Division.
MODE	Selektiert den Anzeige-Modus.
NAME	Benennt eine Datei um.
NEW	Löscht bzw. weist Speicherplatz zu.
ON ADIN GOSUB	Verzweigt bei analogem Interrupt.
ON COMn GOSUB	Verzweigt bei bei seriellem Interrupt.
ON ERROR GOTO	Verzweigt bei Auftritt eines Fehlers.
ON..GOSUB	Fallweiser Unterprogramm-Aufruf.
ON..GOTO	Fallweise Programmverzweigung.
ON KEY GOSUB	Verzweigt bei Tastatur-Interrunpt.
ON PHONE GOSUB	Verzweigt bei Modem-Interrupt.
ON TIME$ GOSUB	Verzweigt bei spezifizierter Uhrzeit.
OPEN	Öffnet Dateien für Zugriffsmöglichkeiten.
OUT	Schreibt Daten zu einem Prozessor-Port.
OUTSTAT	Setzt Steuersignale des RS-232-Interfaces.
PAPER	Setzt Papiertyp und vertikalen Druckbereich.
PASS	Setzt oder löscht ein Kennwort (pass word).
PAUSE	Zeigt Daten für kurze Zeit an.
PCONSOLE	Setzt Format/EOL-Code für Drucker/Interface.
PEEK	Holt Bytes aus Speicherzellen (PC-1600)
PHONE ON/OFF/STOP	Erlaubt/verhindert RS-232C-Interrupts.
PITCH	Bestimmt den Zeichen- u. Zeilenabstand für den Drucker.
POINT	Liefert den Zustand eines Display-Punktes.
POKE	Schreibt Bytes in Speicherzellen (PC-1600).
POWER	Schaltet automatische Abschaltung ein/aus.
PRESET	Löscht einen Display-Punkt.
PRINT	Zeigt Daten auf dem Display an.
PRINT#	Schreibt Daten in eine Datei.
PSET	Setzt oder löscht einen Display-Punkt.
PZONE	Setzt Zone für Drucker oder Interface.
RADIAN	Schaltet Winkelmodus auf Bogenmaß.
RANDOM	Initalisiert den Zufallszahlen-Generator.
RCVSTAT	Bestimmt Empfangsprotokoll und "timeout" (Wartezeit) des seriellen Interfaces.
READ..DATA	Liest Daten aus einer Datenzeile (s.: DATA).
REM	Erlaubt Kommentareinfügungen im Programm.
RENUM	Numeriert die Zeilen eines Programmes neu.
RESTORE	Setzt den DATA-Zeiger (zurück).

Befehl	Kurzbeschreibung
RESUME	Kehrt aus einer Fehler-Routine zurück.
RETURN	Kehrt aus einem Unterprogramm zurück.
RETI	Kehrt aus einer Interrupt-Routine zurück.
RIGHT$	Liefert den rechstbündigen Teil-String.
RLINE	Druckt Gerade in relativen Koordinaten.
RMT ON/OFF	Schaltet die Recorder-Steuerung ein/aus.
RND	Generiert eine Zufallszahl.
ROTATE	Bestimmt die Druckrichtung.
RUN	Startet ein BASIC-Programm.
RXD$	Liefert Daten vom seriellen Interface.
SAVE	Sichert ein BASIC-Programm auf Diskette.
SET	Setzt/Löscht Schreibschutz von Dateien.
SETCOM	Setzt Protokoll für serielles Interface.
SETDEV	Selektiert Interface für Datenausgabe.
SGN	Liefert das Vorzeichen einer Zahl.
SIN	Liefert den Sinus eines Winkelargumentes.
SNDBRK	Sendet Unterbrechungs-Codes über ein serielles Interface aus.
SNDSTAT	Bestimmt Sendeprotokoll und die "timeout" (Wartezeit) des seriellen Interfaces.
SORGN	Setzt Stiftposition als Koordinatenursprung.
STATUS	Liefert Kapazität freier Speicherbereiche.
STOP	Stoppt ein Programm für Korrekturzwecke.
STR$	Wandelt eine Zahl in einen String um.
SQR	Liefert die Quadratwurzel einer Zahl.
TAB	Setzt Druckstift an bestimmte Spalte.
TAN	Liefert den Tangens eines Winkelargumentes.
TEST	Testet alle vier Druckstifte.
TEXT	Setzt den Drucker in den Text-Modus.
TIME	Setzt die Uhrzeit oder gibt sie aus.
TIME$	Liefert einen String der Uhrzeit.
TIME$ ON/OFF/STOP	Erlaubt/verhindert zeitabhängige Interrupts.
TITLE	Selektiert Speicherbereiche.
TROFF	Schaltet den TRACE-Modus aus.
TRON	Schaltet den TRACE-Modus ein.
USING	Sorgt für eine formatierte Datenausgabe.
VAL	Wandelt numerische Strings in Zahlen um.
WAIT	Setzt die Wartezeit für einen PRINT-Befehl.
WAKE$	Setzt Zeit/Befehls-String für automatische Einschaltung des Computers.
XCALL	Ruft Maschinen-Programm auf (PC-1500-Modus).
XPEEK / XPEEK#	Holt Byte aus Speicherzelle (PC-1500-Modus).
XPOKE / XPOKE#	Schreibt Byte in Speicherzelle (" ").

ANHANG H

Kompatibilität zum PC-1500 und dessen Optionen

BASIC-Programme, die auf dem PC-1500 geschrieben worden sind, laufen grundsätzlich auch auf dem PC-1600, sofern dieser im MODE 1 betrieben wird. Die überwiegende Mehrheit der BASIC-Befehle stimmt bei beiden Rechnern überein. Allerdings gibt es auch Abweichungen, die z.B. die standardmäßige Einstellung der Papierbreite des Druckers betreffen. Die in Verbindung mit der Bedienung des Diskettenlaufwerks oder der Handhabung der RAM-Disks stehenden PC-1600-Befehle werden vom PC-1500 nicht unterstützt. PC-1500-Programme, die auf Cassette gespeichert worden sind, können auch vom PC-1600 geladen und ausgeführt werden. Somit ist der Austausch von Programmen zwischen beiden Modellen eigentlich problemlos. Beim Nullsetzen der Uhrzeit ist jedoch zu beachten, daß dieses nicht wie beim PC-1500 durch die Anweisung TIME=0 geschehen kann, da der PC-1600 auch den Monat und den Tag berücksichtigt und diese Daten nicht zu Null werden können. Deshalb muß im Argument immer eine gültige Monats- und Tagesangabe enthalten sein.

Verwendung des CE-150

Wird der PC-1600 zusammen mit dem Drucker CE-150 betrieben, so muß sich der PRINT-Schalter dieses Gerätes in der Punktstellung befinden. Direkte Berechnungen können daher nicht protokolliert werden.

Äquivalente BASIC-Befehle

Die nachfolgende Liste zeigt, welche Änderungen in den Bezeichnungen gleichwertiger Befehle vorgenommen worden ist:

PC-1600	PC-1500	Anmerkung
LLINE	LINE	
LINE		kein Äquivalent
XCALL	CALL	
CALL		kein Äquivalent
XPOKE	POKE	
POKE		kein Äquivalent
XPEEK	PEEK	
PEEK		kein Äquivalent
XPOKE#	POKE#	
XPEEK#	PEEK#	

Weitere Details zur Maschinensprache-Programmierung sind in Anhang E beschrieben.

Emulation des PC-1500

Der PC-1600 kann sich im Betriebsmodus MODE 1 wie ein PC-1500 verhalten, diesen Computer also emulieren. Dabei wird für die Datenausgabe nur die unterste Display-Zeile benutzt. Auch die grafische Ansteuerung des Displays erfolgt wie beim PC-1500. Ebenfalls wird der Zeichensatz dem des PC-1500 angepaßt, so daß in dieser Betriebsart weder die internationalen, die grafischen noch die griechischen Symbole mit den Codes &80 bis &FF genutzt werden können. Den Codes $27 und $5B werden die beim PC-1500 üblichen Zeichen zugeordnet (s. Anhang C).

Tastatureingabe von PC-1500-BASIC-Programmen in den PC-1600

a) Geben Sie (wie gewöhnlich) das Programm im PRO-Modus ein und ändern Sie dabei alle LINE-Befehle in das Äquvalent LLINE.

b) Sichern Sie dieses Programm auf einem beliebigen Medium. Es ist ratsam, in den Kopf des Programmes einen Kommentar auf-aufzunehmen, der Sie daran erinnert, daß es sich hierbei um ein PC-1500-Programm handelt.

Auf Cassette gespeicherte Programme

PC-1500-Programme, die über die Einheit CE-150 und einem daran angeschlossenen Recorder auf Cassette aufgezeichnet worden sind, können im MODE 1 mit Hilfe des CLOAD-Befehles geladen werden, ohne daß diese einer Modifikation unterworfen werden müsssen.

Jedoch können PC-1600-Programme, die man über diese Peripherie auf Cassette aufzeichnet, nicht vom PC-1500 gelesen werden. Dieses ist einerseits durch ein unterschiedliches Aufzeichnungs-verfahren bedingt und andererseits durch die unterschiedlichen Speicherbereiche beider Computer.

Ablauf von PC-1500-BASIC-Programmen auf dem PC-1600

a) Stellen Sie sicher, daß sich ein Modul in einem der beiden Modulfächer befindet. Diese Vorsichtsmaßnahme garantiert, daß ausreichend Speicherplatz zur Verfügung steht. Es sollte hierfür ein Modul mit weniger als 16 KByte Kapazität gwählt werden, also eines der Typen: CE-151, CE-155, CE-159 oder CE-161. Wird kein Modulfach bestückt, hat dieses die Ausgabe eines ERROR-Codes zur Folge.

b) Setzen Sie mit Hilfe des MODE-Befehles den PC-1600 in die Betriebsart MODE 1.

c) Starten Sie dann das Programm wie gewohnt. Alle Resultate sollten dann auf dem Display exakt so erscheinen, wie Sie es vom PC-1500 her kennen.

Drucken mit dem CE-1600P

Alle Druckbefehle des PC-1500, die ursprünglich für den Drucker CE-150 gedacht sind, arbeiten auch im Zusammenhang mit dem neuen Drucker CE-1600P. Wegen der unterschiedlichen Papierbreiten, die diese beiden Drucker verwenden können, ist folgendes zu beachten Soll der Ausdruck des CE-1600P genauso in der Breite formatiert sein wie beim CE-150, muß dieses zuvor mit entsprechenden BASIC-Befehlen vereinbart werden. Diese Befehle können entweder im direkten Modus gestellt oder aber im Programm integriert werden. Sie lauten:

```
CSIZE 2
PZONE "LPT1:",18
PCONSOLE "LPT1:",18,0,0
```

Gebrauch des PC-1600 mit der Einheit CE-150

Der PC-1600 läßt sich, ebenso wie der PC-1500, in die Einheit CE-150 stecken oder mit den Einheiten CE-158 bzw. CE-162 verbinden. Im MODE 1 verhält sich der PC-1600 an diesen Geräten genauso wie ein PC-1500. Dieses gilt auch für den Ausdruck von Programmen, wobei allerdings anstelle der PC-1600-Befehle die Äquivalente des PC-1500 erscheinen. Hierfür ist die in den genannten Geräten enthaltene Software verantwortlich, die nur die PC-1500-Befehle verstehen kann. So erscheint deshalb anstelle des PC-1600-Befehles LLIST das Befehlswort LIST im Ausdruck.

Inkompatibilität zum PC-1500A

Der beim PC-1500A für den Anwender frei benutzbare Speicher im Adreßbereich &7C00 bis &7FFF steht dem Anwender beim PC-1600 nicht mehr zur Verfügung, weil dieser Bereich für Systemzwecke benötigt wird. Aus diesem Grunde können die Modelle PC-1500A und PC-1600 nicht als kompatibel angesehen werden.

Vorsichtsmaßnahmen & Problemlösungshinweise

Ihr SHARP PC-1600 Taschencomputer ist ein hochwertiges Gerät, daß Sie stets mit der nötigen Sorgfalt handhaben sollten, um lange Freude daran zu haben. In den bisherigen Beschreibungen haben wir immer wieder auf Vorsichtsmaßnahmen hingewiesen, die Sie im Umgang mit diesem Gerät beachten sollten, um Probleme zu vermeiden und Gefahren vom Gerät fernzuhalten.

Nachfolgend seien die wichtigsten Punkte noch einmal genannt:

Display Die LCD-Anzeige Ihres PC-1600 besteht aus Glas und ist somit zerbrechlich. Schützen Sie den PC-1600 und sein Display, indem Sie nach Gebrauch dieses Gerätes und insbesondere beim Transport desselben, ihn im mitgelieferten Etui aufbewahren. Vermeiden Sie beim Hantieren mit Verbindungskabel, daß diese gegen das Display schlagen und dieses zerkratzen oder anderweitig beschädigen.

Arbeit und Lagerung Vermeiden Sie, das Gerät extremen Temperaturen oder Feuchtigkeiten auszusetzen. Dieses gilt sowohl für den Gebrauch als auch die Aufbewahrung des PC-1600. Beachten Sie, daß die bei langandauernder Sonnenbestrahlung entstehende Wärme ebenfalls schädlich sein kann. Während trockener Wintermonate besteht bei mit Teppichware ausgelegten Räumen eine besondere Gefahr elektrostatischer Aufladungen. Eine Entladung über das Gerät ist zerstörerisch. Sorgen Sie daher stets für eine Entladung Ihres Körpers an metallischen Gegenständen, bevor Sie mit dem Computer und seinem Zubehör umgehen. Vermeiden Sie in jedem Falle das Berühren von Kontakten und decken Sie diese, sofern sie ungenutzt sind, mit den beiliegenden Kappen ab.

Reinigung Um den Computer und sein Zubehör zu reinigen, sollte ausschließlich ein weiches, trockenes, sauberes und nichtfusselndes Tuch verwendet werden. Benutzen Sie zur Reinigung auf keinen Fall Seife, Lösungsmittel, andere Reiniger oder irgendwelche Flüssigkeiten.

Batterien Wird der Computer für einen längeren Zeitraum nicht genutzt, sollten Sie solange seine Batterien entfernen, um einem Auslauf derselben vorzubeugen und chemische Reaktionen zu vermeiden.

Service Sollte an Ihrem PC-1600 eine Reparatur unerläßlich erscheinen, wenden Sie sich bitte an Ihren SHARP-Händler oder ein von SHARP authorisiertes Service-Center. Damit gehen Sie sicher, daß die Reparaturen sachgemäß durchgeführt werden und vermeiden unnötige zusätzliche Probleme.

Ihr PC-1600 reagiert auf verschiedene Weise, um auf direktem Wege Probleme zu vermeiden oder auf diese alarmierend hinzuweisen. Solche Warnungen erfolgen über die Anzeige in Form von bestimmten ERROR-Codes, deren Bedeutungen Sie anhand der im Anhang F gegebenen Zusammenstellung erkennen können.

Ihr PC-1600 reagiert auf verschiedene Weise, um direkt Probleme zu vermeiden oder auf diese alarmierend hinzuweisen. Warnungen erfolgen über die Anzeige in Form von bestimmten ERROR-Codes, deren Bedeutungen Sie anhand der im Anhang F gegebenen Zusammenstellung erkennen können.

Die nachfolgende Liste gibt Ihnen eine kleine Hilfestellung bei der Behebung der wichtigsten Probleme. Lesen Sie hierzu auch das Kapitel 13, das Ihnen zeigt, wie Probleme, die durch fehlerhafte Programme bedingt sind, beseitigt werden können.

Wir gehen davon aus, daß Sie bei einem Problem zuerst versuchen, dieses selbst zu lokalisieren und zu beheben. Sie werden dabei entdecken, daß Sie durch irrtümlich hervorgerufene Probleme, die Sie selbst aus der Welt schaffen, eine Menge über Ihren PC-1600 lernen, was mit einem Handbuch nicht vermittelt werden kann.

Fehlererscheinung	Folgendes ist zu tun:
Computer ist eingeschaltet, das Display bleibt aber leer	1) Betätigen Sie die OFF -Taste die ON -Taste. Damit ist der Computer mit Sicherheit eingeschaltet. 2) Prüfen Sie, ob auf schwache Batterien oder ungenügend geladenen Akkus hingewiesen wird. 3) Schauen Sie nach, ob der Netzadapter auch tatsächlich richtig angeschlossen ist. 4) Drehen Sie am Kontrast-Einsteller.

Fehlererscheinung	Folgendes ist zu tun:
Das Display ist in Ordnung, aber die Tastatur reagiert nicht	1) Drücken Sie die Taste [CL]. 2) Schalten Sie den PC-1600 aus und dann wieder an. 3) Führen Sie einen einfachen Reset durch Betätigung des RESET-Schalters aus. 4) Halten Sie die [ON]-Taste gedrückt und betätigen Sie zugleich den RESET-Schalter, damit ein Total-Reset ausgeführt wird.
Eingegebene Rechenausdrücke erhalten einen Doppelpunkt nach Eingabe der ersten Zahl	Sie befinden sich versehentlich im PRO-Modus. Drücken Sie auf die [MODE]-Taste, um den RUN-Modus zu aktivieren. Nur hier sind solche Berechnungen erlaubt.

ANHANG J

Technische Spezifikationen

Mikroprozessoren

- Hauptprozessor SC7852 : 3,58 MHz, komatibel zum Z80A
- Nebenprozessor LH5803 : 1,3 MHz, PC-1500-kompatibel
- Hilfsprozessor LU57813P: 307,2 kHz

Display

- Flüssigkristallanzeige mit einstellbarem Kontrast
- Alphanumerische Zeichendarstellung: 4 Zeilen je 26 Zeichen
- Grafische Punktmusterdarstellung : 32 x 156 Punkte

Tastatur

- 65 alphanumerische Tasten
- 6 Funktionstasten (in drei Ebenen programmierbar)

Speicher

- Festwertspeicher (ROM) : 96 KBytes
- Arbeitsspeicher (RAM) : 16 KBytes
- RAM auf ca. 80 KBytes erweiterbar

Für die RAM-Erweiterung sind zwei Modul-Fächer auf der Rückseite des Computers vorgesehen.

Stack-Tiefen

- Für FOR..NEXT-Schleifen : 12 Ebenen
- Für Unterprogramme : 6 Ebenen
- Für Interrupts : 8 Ebenen
- Für schwebende Operationen : 8 Ebenen
- Für schwebende Funktionen : 2 Ebenen

Schnittstellen

- serielle Schnittstelle des Typs RS-232C
- optoelektronische Schnittstelle (SIO) für Glasfaserkabelanschluß
- analoger Dateneingang

Besonderheiten

- Warnung vor verbrauchten Batterien
- Eingebaute Echtzeit-Uhr
- Automatische Ein- und Ausschaltung des Computers
- Automatische Alarmfunktion
- Unterstützung externer Interrupt-Anforderungen
- Kommunikations-Funktionen

Energie-Versorgung und -Bedarf

- 4 Trockenbatterien (Typ SUM-3, AA oder R6) zu je 1,5V
- Anschluß an das Wechselstromnetz über einen Netzadapter möglich

Leistungsaufnahme : 0,48 W
Lebensdauer der Batterien: ca. 25h bei 20°C (68°F) und 10-minütiger Arbeitsdauer sowie 50 minütiger Anzeige verteilt auf eine Betriebsstunde

Physikalische Daten

Gehäuseabmessungen : Länge : 196mm;
Breite: 86mm;
Höhe : 25,5mm
Gewicht (d.h. Masse): ca. 390g inklusive Batterien
Betriebstemperatur : 0°C bis 40°C

Standard-Zubehör

- Aufbewahrungs-Etui
- 2 Tastaturschablonen
- Bedienungshandbuch

Syntax-Diagramme

Die Syntax-Diagramme beschreiben die syntaktischen Regeln, wie sich die Befehle in Verbindung mit ihren zugehörigen Parametern und erforderlichen Trennungszeichen verwenden lassen. Egal, wie komplex die vom Diagramm zu beschreibenden Zusammenhänge auch sein mögen, lassen sie sich alle durch eine geringe Zahl von Grundelementen zusammensetzen.

Diese ovale Umrahmung deutet an, daß es sich bei dem darin abgebildeten Wort um ein Schlüsselwort handelt. Es muß zeichengetreu eingegeben werden. Jeder Buchstabe ist damit, wie in der Abbildung gezeigt, als Großbuchstabe einzutippen.

Beispiel:

Einzugeben ist: LEFT$

Der Kreis symbolisiert ein einfaches Zeichen. Es kann entweder direkt oder aber in Verbindung mit einer Umschalttaste von der Tastatur her zugänglich sein.

Beispiel:

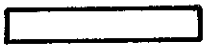
Das kleine Rechteck stellt eine Zusammenfassung von mehreren Einzelzeichen dar. Jedes Zeichen ist getreu der Abbildung einzugegeben.

Beispiel:

Als einzige Ausnahme ist hier der Buchstabe "n" anzusehen, den wir (zur Platzersparnis) bei den beiden Parametern COMn: und Sn: als Stellvertreter für andere Zeichen verwenden.

Beispiel:

Möglich sind: COM1, COM2 oder COM



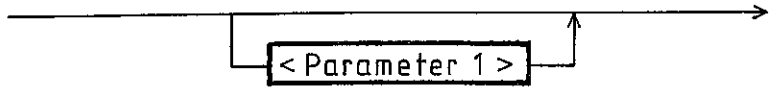
Das große Rechteck enthält Parameter, die nicht nur aus einfachen Zeichen oder Parameterwörtern bestehen können, sondern aus Konstanten, Werten Variablen, Codes usw. Sie sind deshalb nicht in direkter Form angegeben, sondern durch eine Bezeichnung, die in spitzen Klammern eingebettet ist. Diese Klammern sollen darauf hinweisen, daß nicht die Bezeichnung Zeichen für Zeichen einzugegeben ist, sondern der Parameter, der durch diese Bezeichnung beschrieben wird.

Beispiel: <num. Variable>

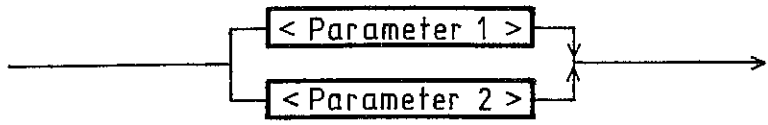
Mögliche Eingaben: N, BETRAG, A(1,3), @(4)

Der Pfeil gibt die Leserichtung an und damit die Folge, in der Schlüsselwort und Parameter aneinandergefügt werden können.

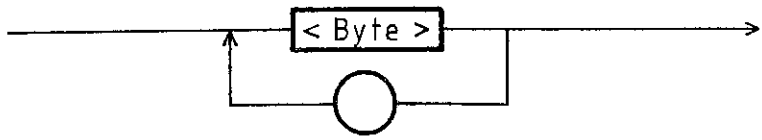
Durch geeignete Anordnung der Pfeile lassen sich auch Alternativen und Aufzählungen darstellen:



Der <Parameter 1> kann angegeben werden, muß es jedoch nicht.



Es kann entweder der <Parameter 1> oder aber der <Parameter 2> dem Befehlswort beigefügt werden.



Dem Befehlswort kann entweder nur ein <Byte> als Parameter beigefügt werden oder aber eine Liste von mehreren Bytes, die jeweils durch ein Komma voneinander zu trennen sind:

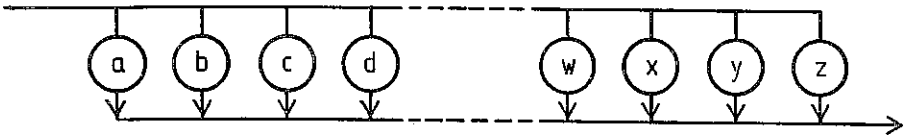
Einzelnes Byte: 255

Liste von Bytes: 107,250,&EA,&9A,147

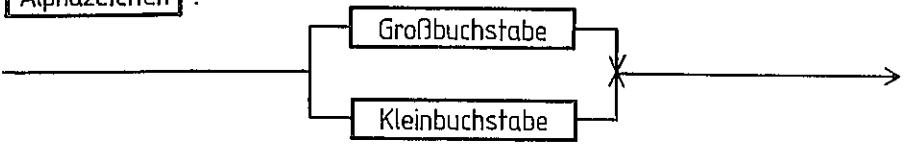
Einteilung der Zeichen

Großbuchstabe :

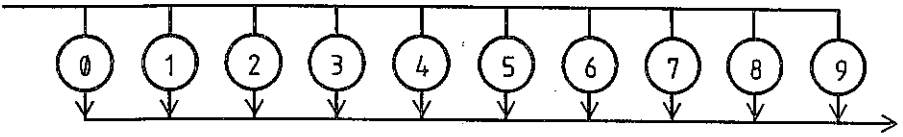
Kleinbuchstabe :



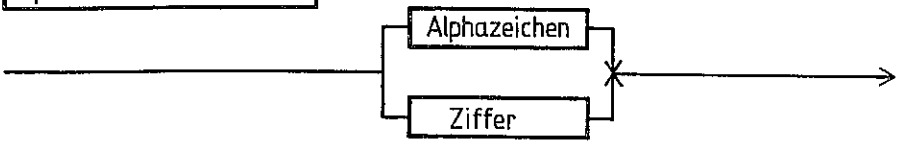
Alphazeichen :



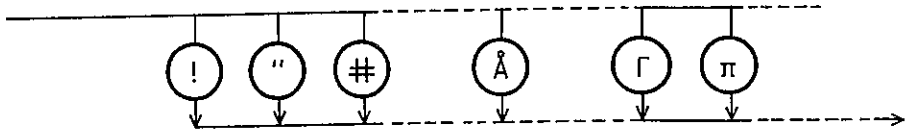
Ziffer :



alphanumerisches Zeichen :

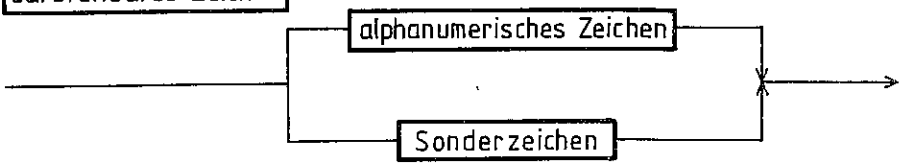


Sonderzeichen :



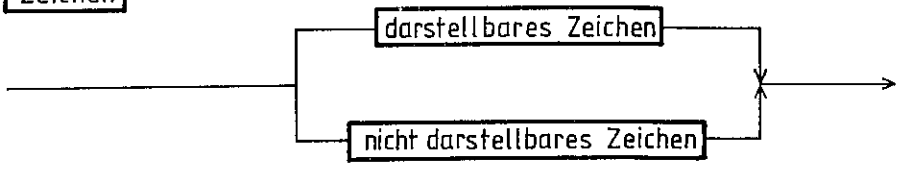
Steuerzeichen :

darstellbares Zeichen :



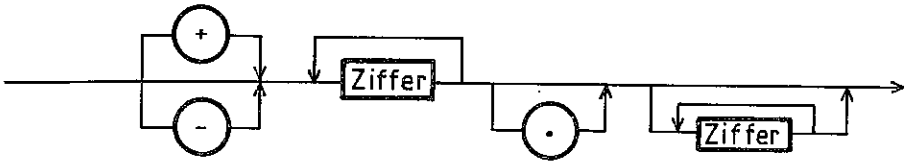
nicht darstellbares Zeichen :

Zeichen :

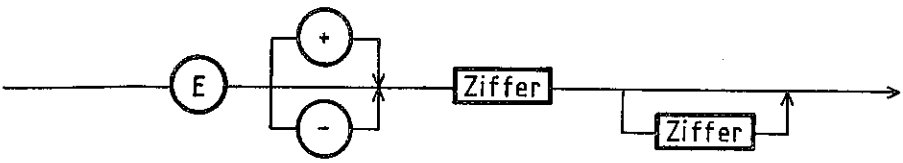


Numerische Werte

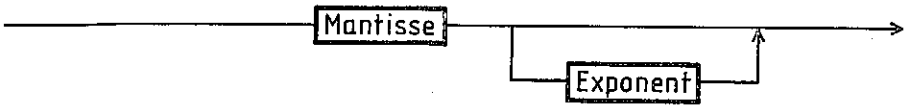
Mantisse :



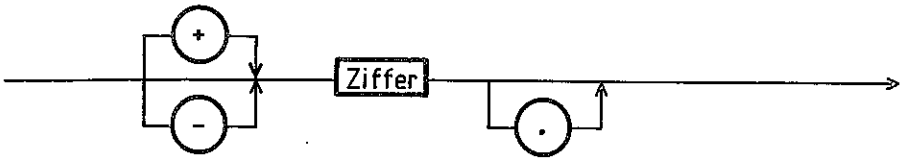
Exponent :



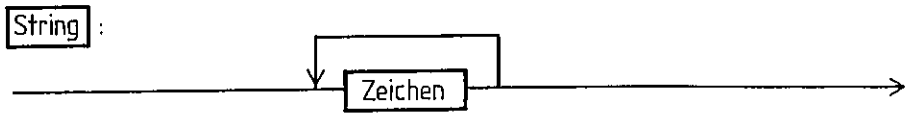
numerische Konstante :

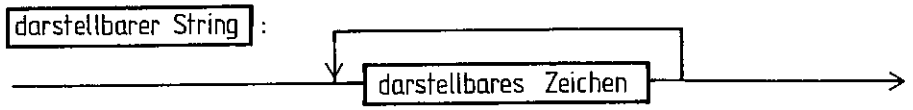


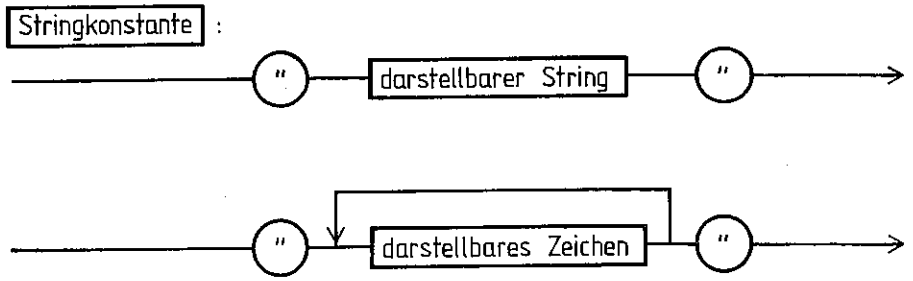
Integer Wert :



Strings

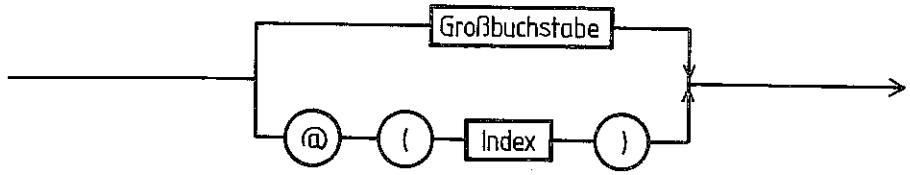

String :
Zeichen


darstellbarer String :
darstellbares Zeichen

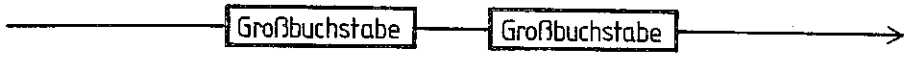

Stringkonstante :
"
darstellbarer String
"
"
darstellbares Zeichen
"

Numerische Variablen

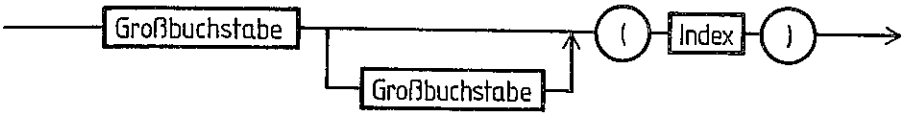
numerische Standardvariable :



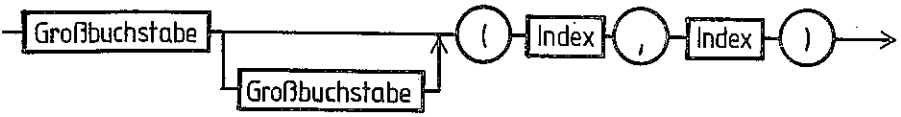
einfache numerische Variable :



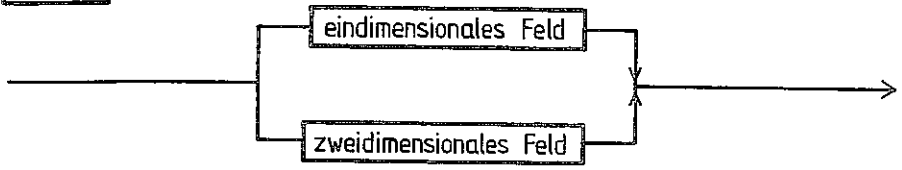
eindimensionales Feld (Array) :



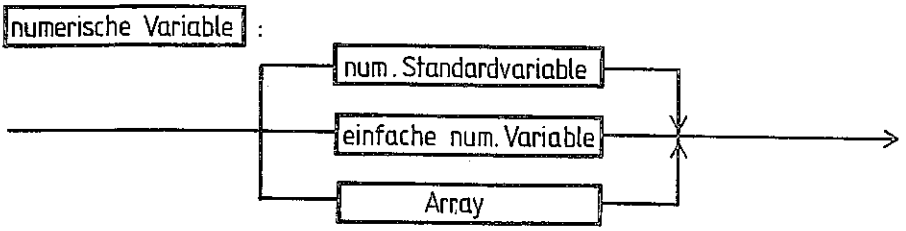
zweidimensionales Feld (Array) :



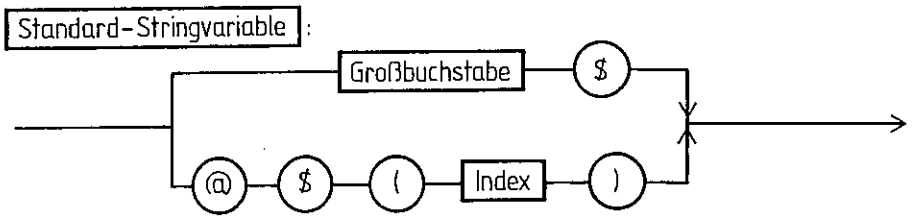
Array :

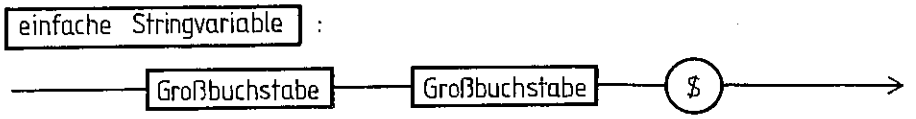


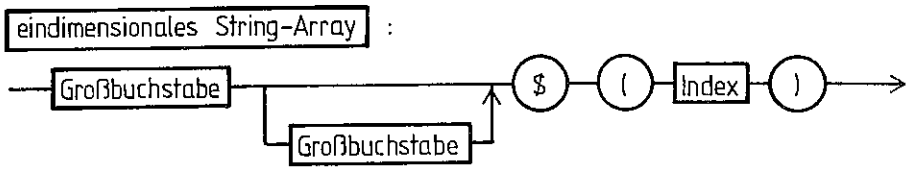
numerische Variable :

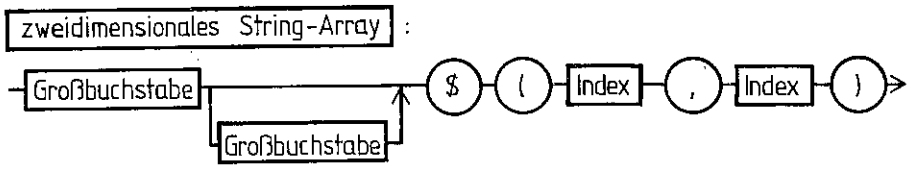


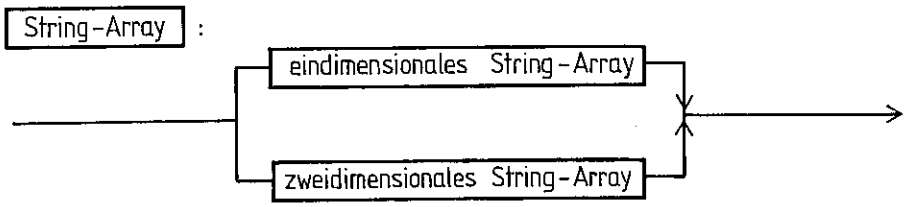
String-Variablen

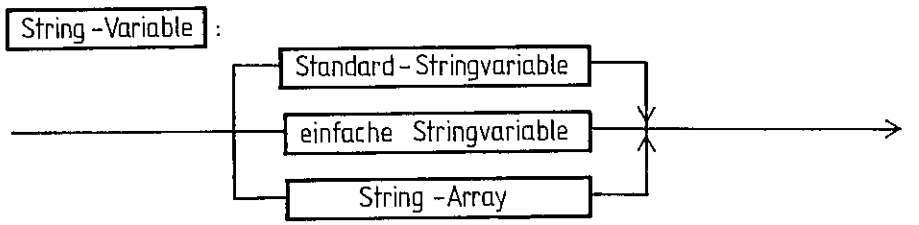








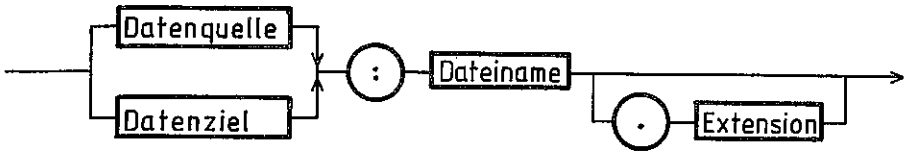




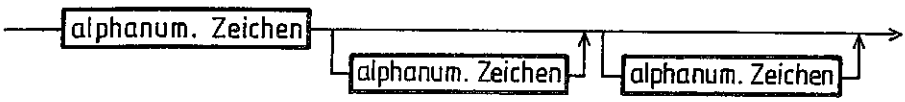
Datei-Bezeichnungen

Dateiname:

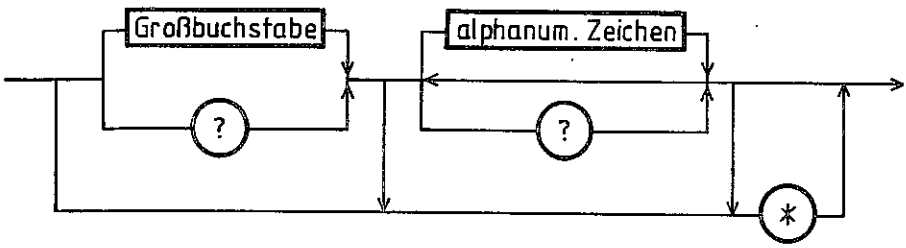
Dateibezeichner:



Extension:



mehrdeutiger Dateiname:



Adapteranschluß 1-3
Änderung einer Datei 11-10
Akkus 3-4, 6-19, 6-20
Algebraischer Ausdruck 10-12
Alphanumerische Tasten 9-1
Alphanumerische Zeichen A-32
Alpha-Tastatur 1-5
Analog-Port 6, 1-2, 6-17
AND-Operator 10-13
Anschluß eines Netzadapters 2-3
Anschlußbelegung
  des RS-232C-Ports 6-11
  des SIO-Interfaces 6-16
Anweisung 14-2
Anwendungssymbolik 14-3
Anzahl der Diskettendateien 11-4
Anzahl der Stop-Bits 12-2
Anzeige 1-2
Anzeige-Modi 4-1, 9-6
Arithmetische BASIC-Funktionen 5-7
Arithmetische Operatoren 10-11
Array
  numerisches Array..... 10-8
  String-Array 10-10
Array-Dimensionierung 10-8
Array-Variablen 10-7
ASCII-Code 10-2, 14-13
Ausdrücke und Operatoren 10-11
Ausgabe über seriellen Port 12-3
Ausschaltung des PC-1600 .....2-3, 3-5
Ausstattungsmerkmale 6
Austausch der RAM-Module A-3
Austausch von Batterien A-1
Automatische Abschaltung 3-5
Auto Power On 3-8
Auto Power Off 3-5, 3-8
Backup Files 6-28
Backspace 4-6
BASIC-Befehle 14-4, A-18
BASIC-Interpreter 1-7
BATT-Symbol 2-2
Batterien
  Einsetzung der Batterien 2-1
  Überwachung der Batterien 2-2
Befehle
  für das Diskettenlaufwerk 6-29
  für den Drucker 6-21
  für den RS-232C-Port 6-12
  für den SIO-Port 6-16

Befehls-Kürzel 14-2
Befehls-Liste A-18
Begriffsdefinitionen 14-1
Benutzer-Bereich 8-6
Bereitschaftszeichen 3-2, 3-3
Betriebsarten 9-1
  Wahl der Betriebsarten 4-4
Binäre Daten 10-1, 10-3
Bit-Muster 10-1, 14-72
Blinken des Cursors 5-5
Boolesche Funktionen 10-13
BREAK-Taste 1-6
BS-Taste 9-4
Byte 10-1
Cassetten-Dateien 11-3
Cassetten-Recorder 6-22
CHECK-Meldung 3-2, 3-3, 3-4, 3-9
CL-Taste 4-6, 9-5
Code 10-2
CTRL-Taste 4-6, 9-2, 9-8
Cursor 3-2
Cursor-Bewegung 4-5
Datei-Bezeichner 11-1, A-38
Dateien 6-29, 11-1
Dateiname 11-1, 11-2, A-38
Dateizugriff 11-9
Datendarstellung 10-1
Datensätze 6-7, 11-1
Datum 4-7, 14-49
Defaults 3-8
DEF-Taste 8-4, 9-3
Device Name ..... s. logischer Gerätename
Direkte Berechnungen 5-1
Direkter Modus 8-1
Dimensionierung 14-53
Disketten-Befehle 6-29
Disketten-Dateien 11-4
Disketten-Formatierung 6-27
Disketten-Kapazität 6-28
Disketten-Laufwerk 6-26
Display 4-1
Display-Koordinaten 9-6, 9-7
Doppelpunkt 8-3
Drucker 6-19
Editier-Modus 4-6, 9-7
Edittierende Tastenfunktionen 4-5
Ein-/Ausgabe-Geräte 6
Einfache Variablen
  einfache numerische Variablen 10-8
  einfache String-Variablen 10-10
Einfacher Reset 3-7, 3-9

Einfüge-Modus 4-6
Einfüge-Symbol 10-3
Einschaltung des PC-1600 3-1
Elektrische Kennwerte des Analog-Ports 6-18
Elementanzahl bei Arrays 10-9
Empfang von Daten und Dateien 12-5
Empfangs-Puffer 12-3
ENTER-Taste 1-6, 5-2, 9-4
Entladene Batterien und Akkus 2-2
ERROR-Codes
  Drucker 6-22
  Disketten-Laufwerk 6-31
  Recorder 6-25
  RAM-Module 6-8
  RS-232C-Port 6-13
  SIO-Port 6-17
Erstellung von Dateien 11-6
Exponent 10-3, 10-5, A-34
Extension 6-28, 11-2, A-38
Externe Geräte 6-1
Fehlerbehandlungs-Routinen 13-2
Fehler-Korrektur 13-1
File Name ..... s. Dateiname
Fixkomma-Konstante 10-4
Fließkomma-Konstante 10-5
Formatierung 6-27, 14-80
Funktionale Operatoren 10-15
Funktions-Tasten 1-6
  Belegung der Funktions-Tasten..... 9-10
Funktionstasten-Menüs 9-12
Funktionstasten-Strings 9-10
Gleichheit 10-12, 14-79
Grafik-Koordinaten 9-6
Grafik-Modus 9-6, 14-76
Grafische-Symbole A-4
Griechische Symbole A-4
Großbuchstabe A-32
Gruppenseparator 11-2
Handshaking 12-2
Hardware-Überblick 1-1
Hexadezimale Konstante 10-5
Hexadezimale Notation 10-1
Hexadezimalziffern 10-1
Hinweise zum Auspacken 1-1
Index 10-6, 10-8
Indirekter Modus 8-1
Inhaltsverzeichnisse 11-4
Insert 4-6, 9-9
Integer-Wert 10-4, A-34
Interfaces 6, 12-1
Internationale Zeichen 9-3

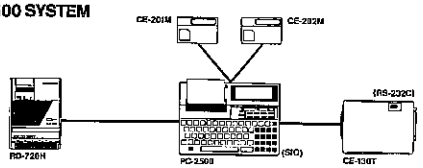
I/O-Devices 6
Joker 6,28, 11-5
Kartoninhalt 1-1
Kaufmansund 10-1, 10-5
Kennwort 11-6, 14-147
Kettenrechnungen 5-4
Kleinbuchstabe A-32
Klick-Funktion 9-3, 14-96
Kommando-Ebene 3-2, 14-1
Kommandos 14-1
Kommunikations-Parameter 12-2
Kommunikations-Protokoll 6-14
Kompatibilität zum PC-1500 ..... 7, A-22
Konstanten 10-4
Kontrast-Einsteller 4-1
Kopieren von Dateien 14-39
Laden von Dateien 11-3, 14-20, 14-30, 14-32, 14-113
Laden von Funktionstasten-Belegungen 9-13
LCD-Display 1-2
Leer-String 10-4
Liste der Befehle A-18
Logische Gerätenamen 11-2
Logische Operatoren 10-13
Logische Zeile 9-7
Mantisse 10-3, A-34
Markierung 8-4
Maschinensprache-Bereich 8-7
Maschinensprache-Programme 1-7
Medium 11-2
Mehrdeutige Datei-Bezeichner 11-5, A-38
Memory Save Guard 3-5
Menü 9-3
  Erzeugung eines Menüs 9-12
  Anzeige eines Menüs 9-12
Menü-Ebenen 9-2
Mignonzellen 2-1
MODE 0 9-6
MODE 1 9-7
MODE-Taste 1-6, 9-1
Modul-Fächer 3-3, 6-3, 6-5
Module 3-3
Netzadapter 2-3
NOT-Operator 10-13
Null-String 10-4
Numerische Daten 10-3
Numerische Felder 10-8
Numerische Konstante 10-4, A-34
Numerische Standard-Variablen 10-7
Numerisches Tastenfeld 1-2
Öffnung einer Datei 11-9
Operatoren 10-11

Optionen 6
Optisches Interface 6-13
OR-Operator 10-13
Overwrite 4-6, 4-7, 9-9
Parität 6-15, 12-2
Password 11-6, 14-147
PC-1500-Computer 7
PC-1500-Optionen 7-1
Peripheriegeräte 6-1
Physikalische Zeile 9-7
Platzhalter 4-6, 9-5
Pocket Disks 6-27
Ports 12-1
Priorität 10-11, 10-1
Problemlösungshinweise A-25
PRO-Modus 4-4, 9-1
Programme 8-1
Programm-Module 3-3, 6-3
Prompt 3-9
Protokoll 6-9, 12-2
Protokoll-Optionen 12-2
Puffer-Batterie 6-3, 6-5
Punktmatrix 6, 9-6
RAM-Disks 3-2, 6-3
RAM-Disk-Befehle 6-6
RAM-Module 6-3, 6-4
  Tabelle der RAM-Module 6-4
RCL-Taste 9-3, 9-12
Rechenbeispiele 5-3
Recorder 6-22
Recorder-Befehle 6-25
Recorder-Interface 6-21, 6-23
Records 6-7, 11-1
RESERVE-Bereich 8-7
RESERVE-Modus 1-6, 4-4, 9-2, 9-9
Reserve Program Area 8-7
Reset-Möglichkeiten 3-6
  Einfacher Reset 3-8, 3-10
  Total-Reset 3-8, 3-9
RESET-Schalter 1-4, 6-20
RS-232C-Befehle 6-12
RS-232C-Interface 6, 1-2, 6-8
Rückholfunktion 5-5
RUN-Modus 4-4, 9-1
Schließen von Dateien 14-33
Schlüsselwörter 14-1
Schnittstellen 6-1
Schreibschutz 6-3, 11-6, 14-188
Schreibschutz-Schalter 6-5, 11-6
Scrolling 5-4
Senden von Daten 12-3

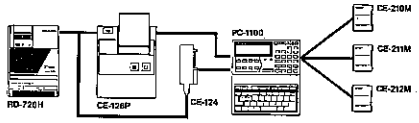
Senden von Dateien 12-4
Senden von Steuer-Codes 12-4
Sequentielle Datei 11-1
Serielle Schnittstellen 6-8
Setzen von Uhrzeit und Datum 4-7
SHARP-fremde Recorder 6-24
SHIFT IN/OUT-Protokoll 12-2
SHIFT-Tasten 1-5, 9-2
SIO 6, 1-2, 6-13
SIO-Befehle 6-16
Sicherheits-Kopien 6-28
SML-Taste 1-5, 9-2
Software, Anmerkungen zur Software 1-7
Speicher-Aufteilung A-6
Speicher-Belegung 8-5
Speicher-Erweiterung 6-3
Speicherkapazität 6-3, 8-6
Speicher-Module 3-3, 6-3
Speicherplatzreservierung 10-6
Speicherschutz 6-4
Speicherung
  von Dateien 14-23, 14-42, 14-43, 14-187
  von Funktionstasten-Belegungen 9-13
Spezifikationen A-28
Spezial-Tasten 9-3
Standard-Variablen
  numerische Variablen 10-7, A-36
  String-Variablen 10-10, A-37
Standardmäßige Einstellungen 3-8
  für RS-232C-Interface 6-10
  für SIO-Interface 6-15
Start eines Programmes 8-3
Status-Zeile 3-2, 4-2
Steuer-Codes 12-4
Stop-Bits 12-2
Stromversorgung 2-1
String 10-3, A-35
String-Arrays 10-10
String-Konstante 10-4, A-35
String-Vergleiche 10-13
Syntax-Diagramme A-30
Syntax-Fehler 13-1
Symbole der Statuszeile 4-2
System-Bereich 8-7
System-Bus 1-3
System-Prompt 3-2
Systemspezifische Einstellungen 3-8
System-Überblick 6-2
Tastatur 1-2, 1-5
Tastenfunktionen 5-2, 9-1
Text-Daten 10-2

Text-Koordinaten 9-6, 9-7
Text-Modus 14-206
Timeout 6-12
Token 9-11
Total-Reset 3-10
Trace-Modus 13-2
Trigonometrische Funktionen 14-41, 14-192, 14-204
Übertragungsprotokoll 6-9, 12-2
Überschreibe-Modus 4-6
Übertragungsgeschwindigkeit 12-2, 12-3
Uhrzeit und Datum 4-8, 4-9
Umschalt-Tasten 9-1
Ungleichheit 10-12, 14-79
User Area 8-6
Variablen 10-5
Variablen-Namen 10-6
Variablen-Typen 10-7
Vergleichsoperatoren 10-12
Versorgungsspannung 6-28
Vordefinierte Tastaturbelegung 9-13
Wildcards 6-28, 11-5
Work Area 8-7
X:-Drive 6-28
XON/XOFF-Protokoll 12-2
Y:-Drive 6-28
Z80-Mikroprozessor 1-7
Zeichen A-32
Zeichen-Codes 10-2
Zeichenkette 4-4, 10-3
Zeilen-Nummer 8-2
Zubehör 6-7
Zugriff auf Dateien 11-9
Zugriff auf die seriellen Ports 12-1
Zweite Tastaturbelegung 9-3

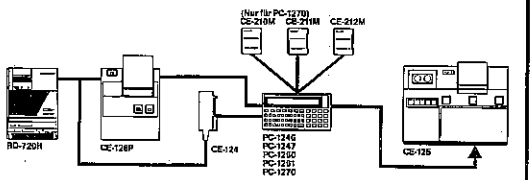
PC-2500 SYSTEM



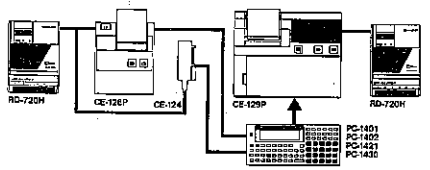
PC-1100 SYSTEM



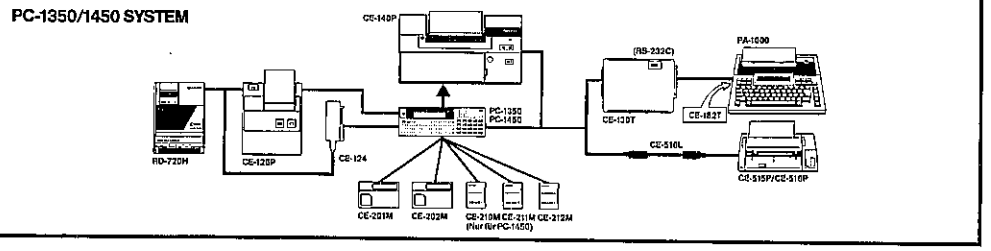
PC-1246/1247/1260/1261/1270 SYSTEM



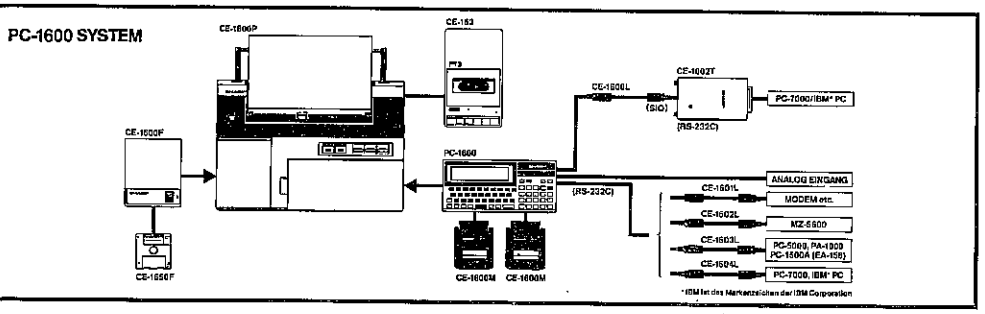
PC-1401/1402/1421/1430 SYSTEM



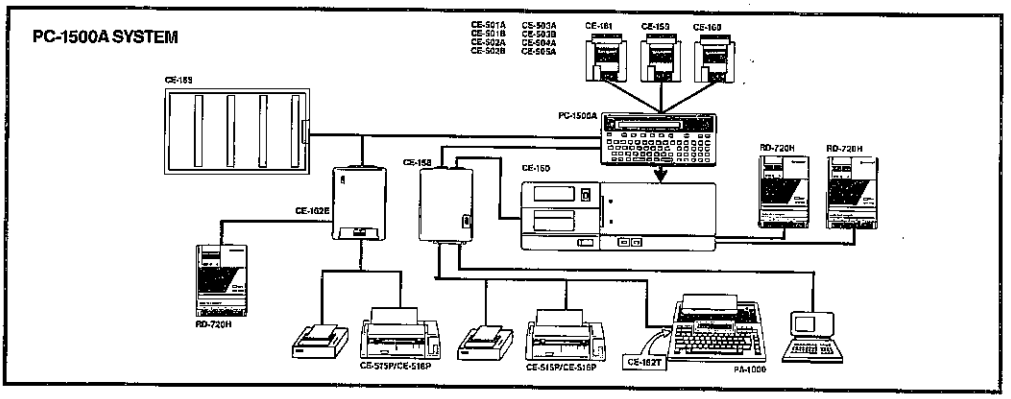
PC-1350/1450 SYSTEM



PC-1600 SYSTEM



PC-1500A SYSTEM



BEZAHLT

23. 12. 91

A. J. KUBERNA

SHARP CORPORATION
OSAKA, JAPAN

PRINTED IN JAPAN/IMPRIMÉ AU JAPON
(TINSG1031EQZZ)